



TUGAS AKHIR - KI141502

Perancangan dan Pembangunan Perangkat Lunak E-QuickCount untuk Pemilihan Umum di Indonesia Berbasis Android

ERLANGGA IBRAHIM NUR WISISONO
NRP 5109 100 169

Dosen Pembimbing
SARWOSRI, S.Kom, MT.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

Design and Building E-QuickCount Software for Indonesian Election based on Android

ERLANGGA IBRAHIM NUR WISISONO
NRP 5109 100 169

Advisor
SARWOSRI, S.Kom, MT.

INFORMATICS ENGINEERING
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji dan syukur, kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“Perancangan dan Pembangunan Perangkat Lunak E-QuickCount untuk Pemilihan Umum di Indonesia Berbasis Android”.

Di lembar ini, penulis ingin menyampaikan penghargaan dan ucapan terima kasih sebesar-besarnya kepada pihak-pihak yang secara langsung maupun tidak langsung telah berkontribusi dalam membantu penyelesaian Tugas Akhir ini. Kepada pihak-pihak tersebut, penulis ingin menyampaikan penghormatan dan terima kasih yang sebesar-besarnya, terutama kepada:

1. Bapak dan Ibuk yang selalu memberikan dukungan dan doa yang tak pernah putus dan tak pernah lelah selama penulis kuliah dan memberikan kekuatan dan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Mbak Nung, mbak Din dan mbak Tin yang selalu mendukung dan membantu penulis baik secara teknis maupun non teknis.
3. Ahet, Abduh, Khurun dan Amung yang telah menemani penulis bermain game di kala stres.
4. Ibu Sarwosri selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
5. Ibu Isye selaku dosen wali yang telah bersedia meluangkan waktu untuk memberikan motivasi selama masa perkuliahan.
6. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.

7. Teman-teman M12 dan Mat'09 yang selalu mengingatkan dan memberikan semangat kepada penulis untuk menyelesaikan Tugas Akhir ini. Kalian bukan teman, tapi sahabat yang luar biasa
8. Suriah Yuli yang selalu memberikan perhatian, canda, tawa dan semangat di sela-sela kesibukan mengerjakan Tugas Akhir.
9. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
10. Teman-teman TC 2009 seperjuangan semasa kuliah atas saran, masukan dan, dukungan terhadap pengerjaan Tugas Akhir ini.
11. Serta semua pihak, teman, keluarga, dan semuanya yang namanya tidak dapat penulis sebutkan satu-persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Januari 2016

Erlangga INW

LEMBAR PENGESAHAN

Perancangan dan Pembangunan Perangkat Lunak E-QuickCount untuk Pemilihan Umum di Indonesia Berbasis Android

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

ERLANGGA IBRAHIM NUR WISISONO

NRP : 5109 100 169

Disetujui oleh Dosen Pembimbing Tugas Akhir :

SARWOSRI, S.Kom
NIP: 19760809 200112 0001 (pembimbing)



**SURABAYA
JANUARI 2016**

[Halaman ini sengaja dikosongkan]

Perancangan dan Pembangunan Perangkat Lunak E-QuickCount untuk Pemilihan Umum di Indonesia Berbasis Android.

Nama Mahasiswa : ERLANGGA IBRAHIM NUR WISISONO
NRP : 5109 100 169
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing : SARWOSRI, S.Kom, MT

ABSTRAK

Quick count merupakan proses perhitungan cepat hasil dari suatu voting atau dewasa ini sering dikaitkan dengan proses perhitungan cepat Hasil Pemilu di Indonesia. Selama ini, teknologi yang digunakan oleh Lembaga Survei berupa Teknologi SMS (Short Message Service) yang didukung program SMS-Gateway dan pengolahan datanya secara manual oleh para admin di kantor pusat data. Teknologi ini memiliki beberapa kekurangan, yaitu dibutuhkan banyak SDM di kantor pusat data, belum lagi SDM yang diterjunkan ke lapangan atau TPS sampel. Karena proses pengolahannya masih manual sehingga dibutuhkan jeda waktu atau waktu lebih dari selesainya proses menghitung di TPS hingga penampilan hasil perhitungannya di media.

Tugas akhir ini membangun aplikasi e-QuickCount, dimana aplikasi ini akan terinstal pada perangkat bergerak dengan platform android. Aplikasi ini akan digunakan oleh petugas di TPS sampel untuk memasukkan nilai perolehan suara pasangan calon lalu langsung mengunggahnya untuk diolah di web service. Data hasil olahan web service akan langsung ditampilkan di aplikasi web dalam bentuk grafik dan tabel,

sehingga tidak ada waktu yang terbuang dan hasil bisa langsung segera diketahui oleh masyarakat umum.

Dengan diterapkannya aplikasi e-QuickCount ini pada proses perhitungan suara Pemilu yang ada di Indonesia, waktu perhitungan tersebut bisa dipangkas habis, karena data unggahan dari petugas langsung diolah di web service dan langsung ditampilkan juga ke aplikasi web. Karena semua proses diolah oleh komputer, maka jumlah SDM dapat ditekan hanya untuk petugas lapangan dan admin aplikasi web saja, sehingga baik waktu maupun biaya bisa lebih diminimalkan.

Kata kunci: Pemilu, Quick Count, TPS, Web Service, Android.

Design and Building E-QuickCount Software for Indonesian Election based on Android

Student Name : Erlangga Ibrahim Nur Wisisono
NRP : 5109 100 169
Major : Informatics Engineering FTIf-ITS
Advisor : SARWOSRI, S.Kom, MT

ABSTRACT

Quick count is a quick calculation process or the outcome of today's voting is often associated with rapid calculation process election results in Indonesia. During this time, the technology used by the Survey Institute is SMS (Short Message Service) technology supported with SMS-Gateway program and processing the data manually by the employee in data center office. This technology has some shortcomings, which required a lot of human resources at the headquarters of the data, not to mention the human resources deployed to the field or TPS sample. Due to the processing process is still manual so it takes a lag time or more time than the completion of the process of counting at the polling stations until the appearance of the calculation results in the media.

The final task is to build e-QuickCount applications, where the application will be installed on mobile devices with android platform. This application will be used by officers in the polling sample to include the value of the candidate's vote then immediately upload them to be processed in the web service. Data processed by the web service will be directly displayed in graphic and table in a web application, so no time is wasted, and the results can be directly immediately known by the general public.

With the implementation of e-QuickCount apps in the process of counting the election in Indonesia, the calculation time can be cut off, because the data uploads from the clerk processed directly on the web service and immediately shown also to web applications. Because all processes processed by a computer, then the number of Human Resource can be suppressed only for field officers and administrators of web applications, so that both time and costs can be minimalized.

Keyword: *Election, Quick Count, TPS, Web Service, Android.*

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER.....	xxvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	3
1.3. Rumusan Permasalahan	3
1.4. Batasan Permasalahan.....	4
1.5. Manfaat Tugas Akhir	4
1.6. Metodologi.....	4
1.7. Sistematika Penulisan	7
BAB II DASAR TEORI.....	9
2.1. Penelitian Terkait	9
2.2. Pemilihan Umum	10
2.3. Kepala Daerah.....	11
2.4. Quick Count.....	11
2.5. Metodologi Quick Count	12
2.6. Penarikan Sampel Quick Count	12
2.7. Web Service	12

2.8.	Android	14
2.9.	Eclipse	14
2.10.	Adobe Dreamweaver	14
BAB III ANALISIS DAN PERANCANGAN		17
3.1.	Analisis	17
3.1.1.	Domain Permasalahan.....	17
3.1.2.	Deskripsi Umum Perangkat Lunak	17
3.1.3.	Arsitektur Perangkat Lunak	18
3.1.4.	Spesifikasi Kebutuhan Perangkat Lunak.....	18
3.1.5.	Analisa <i>Stakeholder</i>	19
3.1.6.	Skenario Kasus Penggunaan	19
3.1.6.1.	Kasus Pengguna Melakukan Login di Aplikasi Mobile23	
3.1.6.2.	Kasus Pengguna Memasukkan Hasil Perhitungan Suara	24
3.1.6.3.	Kasus Pengguna Mencari Kota atau Provinsi Tempat Pemilu Berlangsung	26
3.1.6.4.	Kasus Pengguna Melakukan Login di Aplikasi Web .	27
3.1.6.5.	Kasus Pengguna Melakukan Registrasi Pemilu Baru	29
3.1.6.6.	Kasus Pengguna Melakukan Registrasi Petugas TPS Sampel	31
3.2.	Tahap Perancangan	32
3.2.1.	Perancangan Data	32
3.2.3.1	Tabel admin.....	34
3.2.3.2	Tabel calon.....	35
3.2.3.3	Tabel info_peserta	35
3.2.3.4	Tabel tps	35

3.2.3.5	Tabel kota.....	36
3.2.1.5	Tabel list pemilu	36
3.2.1.6	Tabel pemilu_kota	36
3.2.1.7	Tabel pemilu_provinsi	37
3.2.1.8	Tabel petugas	37
3.2.1.9	Tabel provinsi	38
3.2.2.	Perancangan Antarmuka Grafis	38
3.2.2.1	Rancangan Antarmuka Halaman Pembuka Mobile.	38
3.2.2.2	Rancangan Antarmuka Halaman Login Petugas.....	39
3.2.2.3	Rancangan Antarmuka Halaman Unggah Nilai Suara Pasangan Calon	39
3.2.2.4	Rancangan Antarmuka Halaman Home Web	40
3.2.2.5	Rancangan Antarmuka Halaman Pemilu Kota/Kabupaten	41
3.2.2.6	Rancangan Antarmuka Halaman Pemilu Provinsi ...	42
3.2.2.7	Rancangan Antarmuka Halaman Grafik Pemilu Kota/Kabupaten	44
3.2.2.8	Rancangan Antarmuka Halaman Grafik Pemilu Provinsi	44
3.2.2.9	Rancangan Antarmuka Halaman Login Admin Web	45
3.2.2.10	Rancangan Antarmuka Halaman Pendaftaran Pemilu	46
3.2.2.11	Rancangan Antarmuka Halaman Pendaftaran Petugas	46
3.2.3.	Perancangan Webservice	47
3.2.3.1.	Login Aplikasi Mobile	47
3.2.3.2.	Menampilkan Keterangan dan Calon Pemilu	47

3.2.3.3.	Update Perolehan Suara	48
BAB IV IMPLEMENTASI.....		49
4.1.	Lingkungan Implementasi.....	49
4.2.	Implementasi Database	49
4.3.	Implementasi Antar Muka.....	51
4.4.	Implementasi Arsitektur.....	53
4.5.	Implementasi Melakukan Login Aplikasi Mobile ...	54
4.6.	Implementasi Menampilkan Keterangan Pemilu.....	54
4.7.	Implementasi Unggah Data Nilai Suara Pasangan Calon.....	55
4.8.	Implementasi Memilih Tempat Pemilu	56
4.9.	Implementasi Menampilkan Grafik Pemilu Kota/Kabupaten	57
4.10.	Implementasi Menampilkan Grafik Pemilu provinsi	58
4.11.	Implementasi Menghitung Prosentase Nilai Suara yang Masuk.....	59
4.12.	Implementasi Login Admin Aplikasi Web.....	60
4.13.	Implementasi Mendaftar Pemilu Baru.....	61
4.14.	Implementasi Mendaftar Petugas	61
BAB V PENGUJIAN DAN EVALUASI.....		63
5.1.	Lingkungan Pelaksanaan Pengujian	63
5.1.1.	Lingkungan Perangkat Ponsel Pintar	63
5.1.2.	Lingkungan Perangkat Lunak	64
5.2.	Dasar Pengujian	64
5.3.	Skenario Pengujian	65
5.3.1	Pengujian Fungsional	65

5.3.1.1.	Pengujian Melakukan Login Aplikasi Mobile	65
5.3.1.2.	Pengujian Mengunggah Nilai Suara Calon.....	66
5.3.1.3.	Pengujian Melakukan Login Admin Aplikasi Web ...	67
5.3.1.4.	Pengujian Mendaftarkan Pemilu Gubernur	68
5.3.1.5.	Pengujian Mendaftarkan Pemilu Walikota/Bupati..	71
5.3.1.6.	Pengujian Mendaftarkan Petugas TPS Sampel.....	73
5.3.1.7.	Pengujian Memilih Tempat Pemilu Gubernur	74
5.3.1.8.	Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Gubernur.....	75
5.3.1.9.	Pengujian Memilih Tempat Pemilu Walikota/Bupati	77
5.3.1.10.	Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Walikota/Bupati	78
5.4	Evaluasi Pengujian.....	79
5.5	Pengujian Langsung.....	81
BAB VI PENUTUP		85
6.1.	Kesimpulan	85
6.2.	Saran	86
DAFTAR PUSTAKA		87
LAMPIRAN A IMPLEMENTASI.....		89
BIODATA PENULIS		115

[HALAMAN INI SENGAJA DIKOSONGKAN]

DAFTAR TABEL

Tabel 3.1	Deskripsi Kasus Penggunaan.....	22
Tabel 3.2	Skenario Kasus Pengguna: Melakukan Login di Aplikasi Mobile	23
Tabel 3.3	Skenario Kasus Pengguna: Memasukkan Hasil Perhitungan Suara	24
Tabel 3.4	Skenario Kasus Pengguna: Mencari Kota Atau Provinsi Tempat Pemilu Berlangsung	26
Tabel 3.5	Skenario Kasus Pengguna: Melakukan Login Di Aplikasi Web	28
Tabel 3.6	Skenario Kasus Pengguna: Melakukan Registrasi Pemilu Baru	29
Tabel 3.7	Skenario Kasus Pengguna: Melakukan Registrasi Petugas TPS Sampel	31
Tabel 3.8	Tabel admin.....	35
Tabel 3.9	Tabel calon.....	35
Tabel 3.10	Tabel info_peserta	35
Tabel 3.11	Tabel tps	36
Tabel 3.12	Tabel kota	36
Tabel 3.13	Tabel list_pemilu.....	36
Tabel 3.14	Tabel pemilu_kota.....	37
Tabel 3.15	Tabel pemilu_provinsi	37
Tabel 3.16	Tabel petugas.....	37
Tabel 3.17	Tabel provinsi.....	38
Tabel 5.1	Lingkungan Perangkat Keras.....	63
Tabel 5.2	Lingkungan Perangkat Lunak.....	64
Tabel 5.3	Pengujian Melakukan Login Aplikasi Mobile	65
Tabel 5.4	Pengujian Mengunggah Nilai Suara Calon	66
Tabel 5.5	Pengujian Melakukan Login Admin Aplikasi Web ..	67
Tabel 5.6	Pengujian Mendaftarkan Pemilu Gubernur.....	69
Tabel 5.7	Pengujian Mendaftarkan Pemilu Walikota/Bupati....	71
Tabel 5.8	Pengujian Mendaftarkan Petugas TPS Sampel	73
Tabel 5.9	Pengujian Memilih Tempat Pemilu Gubernur	74

Tabel 5.10 Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Gubernur	75
Tabel 5.11 Pengujian Memilih Tempat Pemilu Walikota/Bupati	77
Tabel 5.12 Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Walikota/Bupati	78

DAFTAR GAMBAR

Gambar 1.1	Tahap Pengerjaan Tugas Akhir	5
Gambar 2.1	Deskripsi Umum Aplikasi	10
Gambar 2.2	Fungsi dari Web Service.....	13
Gambar 3.1	Arsitektur Sistem.....	18
Gambar 3.2	Diagram Kasus Penggunaan	20
Gambar 3.3	Diagram Aktivitas Sistem.....	21
Gambar 3.4	Diagram Aktivitas Melakukan Login Aplikasi Mobile	24
Gambar 3.5	Diagram Aktivitas Memasukkan Hasil Perhitungan Suara.....	25
Gambar 3.6	Diagram Aktivitas Mencari Kota Atau Provinsi Tempat Pemilu Berlangsung.....	27
Gambar 3.7	Diagram Aktivitas Melakukan Login Di Aplikasi Web	28
Gambar 3.8	Diagram Aktivitas Melakukan Registrasi Pemilu Baru.....	30
Gambar 3.9	Diagram Aktivitas Melakukan Registrasi Petugas Tps Sampel.....	32
Gambar 3.10	CDM.....	33
Gambar 3.11	PDM.....	34
Gambar 3.12	Rancangan Antarmuka Halaman Pembuka Mobile	38
Gambar 3.13	Rancangan Antarmuka Halaman Login Petugas..	39
Gambar 3.14	Rancangan Antarmuka Unggah Nilai Suara Pasangan Calon	40
Gambar 3.15	Rancangan Antarmuka Halaman Home Web	41
Gambar 3.16	Rancangan Antarmuka Halaman Pemilu Kota/Kabupaten.....	42
Gambar 3.17	Rancangan Antarmuka Halaman Pemilu Provinsi	43
Gambar 3.18	Rancangan Antarmuka Halaman <i>Grafik Pemilu</i> <i>Kota/Kabupaten</i>	43
Gambar 3.19	Rancangan Antarmuka Halaman Grafik Pemilu Provinsi	44

Gambar 3.20	Rancangan Antarmuka Halaman <i>Login Admin Web</i>	45
Gambar 3.21	Rancangan Antarmuka Halaman Pendaftaran Pemilu	45
Gambar 3.22	Rancangan Antarmuka Halaman Pendaftaran Petugas	46
Gambar 4.1	Antar Muka Halaman Login dan Unggah Nilai ...	53
Gambar 4.2	Implementasi Melakukan Login Aplikasi Mobile	54
Gambar 4.3	Implementasi Menampilkan Keterangan Pemilu	55
Gambar 4.4	Implementasi Unggah Nilai	56
Gambar 4.5	Implementasi Memilih Tempat Pemilu	57
Gambar 4.6	Implementasi Menampilkan Grafik Pemilu Kota/Kabupaten.....	58
Gambar 4.7	Implementasi Menampilkan Grafik Pemilu Provinsi	59
Gambar 4.8	Implementasi Menghitung Prosentase Nilai Suara Masuk.....	60
Gambar 4.9	Implementasi Login Admin Aplikasi Web	60
Gambar 4.10	Implementasi Mendaftar Pemilu Baru	61
Gambar 4.11	Implementasi Mendaftar Petugas.....	62
Gambar 5.1	Pengujian Login Aplikasi Mobile.....	66
Gambar 5.2	Pengujian Mengunggah Nilai	67
Gambar 5.3	Pengujian Login Admin Web	68
Gambar 5.4	Pesan Data Pemilu Gubernur Berhasil Didaftarkan	69
Gambar 5.5	Pengujian Mendaftarkan Pemilu Gubernur.....	70
Gambar 5.6	Pengujian Mendaftarkan Jumlah TPS Pemilu Gubernur	70
Gambar 5.7	Pesan Data Jumlah TPS Pemilu Gubernur Berhasil Dimasukkan.....	70
Gambar 5.8	Pengujian Mendaftarkan Pemilu Walikota/Bupati	72
Gambar 5.9	Pesan Data Pemilu Walikota/Bupati Berhasil Didaftarkan.....	72
Gambar 5.10	Pesan Data Petugas TPS Sampel Berhasil Didaftarkan.....	73

Gambar 5.11	Pengujian Mendaftarkan Petugas TPS Sampel	74
Gambar 5.12	Melihat Grafik Hasil Perhitungan Pemilu Gubernur	76
Gambar 5.13	Melihat Tabel Hasil Perhitungan Pemilu Gubernur	76
Gambar 5.14	Pengujian Melihat Grafik Hasil Perhitungan Pemilu Walikota/Bupati	79
Gambar 5.15	Kondisi TPS Keputih Gang Pasar.....	82
Gambar 5.16	Kondisi TPS Perumdos Blok J	82
Gambar 5.17	Kondisi TPS Perumdos Blok T.....	83
Gambar 5.18	Kondisi TPS Perumdos Blok U	83
Gambar 5.19	Implementasi Login dan Isi Data.....	84
Gambar 5.20	Implementasi Unggah Data	84

[HALAMAN INI SENGAJA DIKOSONGKAN]

DAFTAR KODE SUMBER

Kode Sumber A. 1	Implementasi Pertukaran Data	89
Kode Sumber A. 2	Implementasi Login Aplikasi Mobile	97
Kode Sumber A. 3	Implementasi Menampilkan Keterangan Pemilu	99
Kode Sumber A. 4	Implementasi Unggah Data Nilai Suara Calon.....	101
Kode Sumber A. 5	Implementasi Memilih Tempat Pemilu	103
Kode Sumber A. 6	Implementasi Menampilkan Grafik Pemilu Kabupaten	105
Kode Sumber A. 7	Implementasi Menampilkan Grafik Pemilu Provinsi	107
Kode Sumber A. 8	Implementasi Menghitung Prosentase Nilai	109
Kode Sumber A. 9	Implementasi Login Admin Aplikasi Web	110
Kode Sumber A. 10	Implementasi Mendaftar Pemilu Baru	111
Kode Sumber A. 11	Implementasi Mendaftar Petugas TPS	113

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Kondisi dunia IT saat ini telah berkembang menjadi semakin kompleks, semakin kompetitif, bergerak dengan cepat serta semakin sulit untuk diprediksi. Agar dapat bersaing dan sukses, setiap individu ataupun organisasi perlu menyesuaikan diri dengan segala perkembangan yang ada, termasuk Lembaga Perhitungan Cepat atau quick count. Selama ini, teknologi yang dipakai berupa short message service (SMS) yang didukung dengan program SMS-gateway, serta pengolahan secara manual data suara para calon menggunakan Microsoft Excel di kantor pusat data.

Pelaporan hasil perhitungan suara oleh petugas TPS sampel dalam proses quick count menggunakan teknologi SMS. Proses diawali dengan petugas TPS sampel mengirim SMS yang berisi laporan hasil perhitungan suara ke admin yang berada di kantor pusat data. Data dari petugas tersebut dicatat dan diolah oleh para admin menggunakan Microsoft Excel. Teknologi yang dipakai saat ini memiliki kelebihan serta kekurangan.

Kelebihannya yaitu tidak diperlukan pelatihan khusus karena setiap petugas dapat langsung menggunakan SMS. Kekurangannya dibutuhkan waktu yang cukup lama untuk mengetahui hasil akhir perhitungan cepat, dikarenakan waktu yang dibutuhkan di setiap TPS sampel untuk menyelesaikan proses perhitungan suara berbeda-beda, ada yang cepat ada

yang lambat padahal para admin di kantor pusat data setelah mendapatkan SMS dari seluruh petugas di setiap TPS lapangan masih membutuhkan waktu untuk mengolah data-data tersebut sampai dapat menampilkannya menjadi hasil akhir quick count. Kelemahan selanjutnya dibutuhkan banyak sumber daya di kantor pusat data.

Dari kelemahan teknologi yang digunakan dalam proses perhitungan cepat atau quick count tersebut maka dibangunlah aplikasi e-QuickCount. Aplikasi ini nantinya akan digunakan oleh petugas di TPS sampel. Serta dibuat sebuah web service yang mendukung e-QuickCount ini dalam hal mengolah semua proses quick count, mulai dari mengunggah data perolehan suara para calon sampai menampilkan hasil akhir perhitungan cepat atau quick count di aplikasi web yang sudah disediakan.

Langkah-langkah dalam aplikasi e-QuickCount ini dimulai dari petugas TPS sampel memasukkan data hasil perhitungan suara Calon Kepala Daerah melalui e-QuickCount. Setelah entri data, petugas mengunggahnya dengan menggunakan jaringan internet ke web service. Data unggahan dari seluruh petugas di setiap TPS sampel ini akan mengalami proses pengolahan serta perhitungan di dalam web service dan selanjutnya hasil perhitungan tersebut akan langsung ditampilkan dalam bentuk grafik di aplikasi web. Seluruh masyarakat dapat melihat atau memantau maupun mengunduh langsung hasil perhitungan cepat ini melalui aplikasi web. Dengan menambahkan sistem komputer pada proses perhitungan cepat atau quick count Pemilu melalui penggunaan aplikasi e-QuickCount ini diharapkan dapat mempercepat proses quick count dan menghemat sumber daya yang ada.

1.2. Tujuan

Tugas Akhir ini memiliki tujuan yang rinciannya dapat dituliskan sebagai berikut.

1. Membuat aplikasi berbasis android untuk petugas TPS sampel dalam proses perhitungan cepat atau quick count Pemilukada di Indonesia.
2. Membuat aplikasi web untuk menampilkan informasi hasil perhitungan cepat atau quick count Pemilukada di Indonesia.
3. Membuat sebuah web service untuk menangani proses perhitungan cepat atau quick count mulai dari perhitungan suara di TPS sampel hingga mengolah data suara sampai keluar grafik hasil perhitungannya.
4. Mempercepat waktu dalam proses perhitungan cepat atau quick count.

1.3. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut.

1. Bagaimana membuat aplikasi berbasis android untuk petugas TPS sampel dalam proses perhitungan cepat atau quick count Pemilu di Indonesia?
2. Bagaimana membuat aplikasi web untuk menampilkan informasi hasil perhitungan cepat atau quick count Pemilukada di Indonesia.
3. Bagaimana membuat sebuah web service untuk menangani proses perhitungan cepat atau quick count mulai dari perhitungan suara di TPS sampel hingga mengolah data suara sampai keluar grafik hasil perhitungannya?
4. Bagaimana mempercepat waktu dalam proses perhitungan cepat atau quick count?

1.4. Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut.

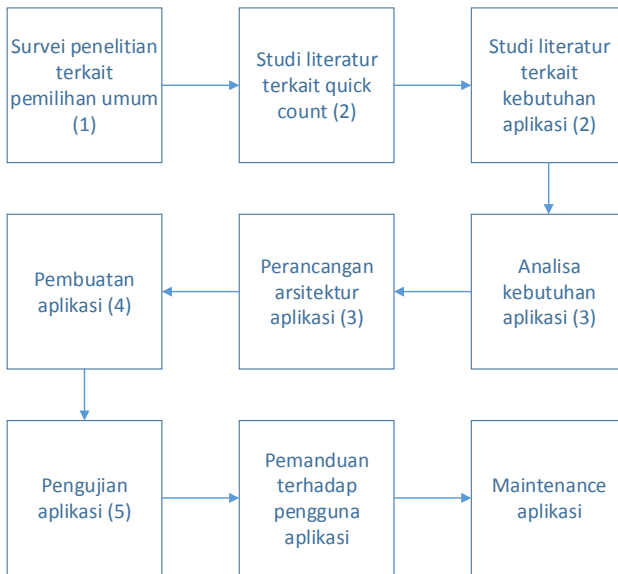
1. Aplikasi mobile yang akan dibangun hanya dapat dijalankan pada alat komunikasi bergerak yang memakai sistem operasi Android.
2. Pemilihan TPS sampel Pemilu tidak melalui aplikasi ini.
3. Aplikasi yang dibuat hanya untuk Pemilu Kota/Kabupaten dan Pemilu Provinsi.
4. Pengembangan aplikasi ini memakai bahasa pemrograman java dan PHP.
5. Ruang lingkup aplikasi hanya untuk proses quick count Pemilihan Umum Kepala Daerah di Indonesia.

1.5. Manfaat Tugas Akhir

Manfaat dari diselesaikannya tugas akhir ini adalah membantu pihak Lembaga Survei dalam mempercepat proses perhitungan cepat atau quick count Pemilu yang sedang berlangsung.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini adalah sebagai berikut ini.



Gambar 1.1 Tahap Pengerjaan Tugas Akhir

1. Penelitian Terkait

Yang pertama kali dilakukan yaitu mencari penelitian terkait dengan judul dari Tugas Akhir ini, yaitu mencari penelitian yang berkaitan tentang pemilu, quick count perhitungan suara dalam pemilu, perhitungan nyata atau real count pemilu dll. Penelitian ini dapat dicari dari berbagai sumber, seperti dari paper, buku, majalah maupun dari internet.

2. Analisis Kebutuhan dan Studi Literatur

Dalam proses selanjutnya yaitu tahap studi literatur tentang semua hal yang ada hubungannya dengan pengerjaan tugas akhir. Studi tentang metode quick count, android, aplikasi bergerak, dan web service. Literatur yang digunakan dalam pengerjaan tugas akhir ini

sebagian besar berasal dari internet yang berupa ebook tutorial, artikel, materi kuliah, serta beberapa buku referensi.

3. Perancangan Sistem

Proses perancangan dimulai dengan mempelajari metode quick count yang sedang dipakai oleh Lembaga Survei sekarang ini, mekanismenya, kelemahan serta kelebihanannya. Setelah itu mulai merancang metode perhitungan cepat atau quick count yang lebih praktis dan efisien, dimulai dari menganalisa kebutuhan, merancang diagram kasus penggunaan, dan merancang arsitektur sistem dan perangkat lunak.

4. Pembuatan (Implementasi)

Untuk rencana pengembangan aplikasi android ini nantinya akan menggunakan Eclipse sebagai IDE nya. Untuk aplikasi web dan webservice nya menggunakan bahasa PHP dalam proses pembuatannya.

5. Pengujian dan Evaluasi

Pada tahap ini aplikasi di uji coba dengan cara dibuat suatu simulasi pemilihan umum, dimana beberapa mahasiswa berperan sebagai petugas TPS sampel yang melakukan proses pencatatan hasil perhitungan suara melalui e-QuickCount dan mengirimnya untuk diolah di webservice. Pada proses ini kinerja sistem diamati dan dicatat segala hal yang terjadi untuk dilakukan evaluasi kedepannya.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir. Buku Tugas

Akhir ini bertujuan untuk memberikan gambaran dari pengerjaan Tugas Akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

1.7. Sistematika Penulisan

Pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir. Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

Secara garis besar, buku Tugas Akhir nantinya terdiri atas beberapa bagian, yaitu sebagai berikut.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, manfaat, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Tinjauan Pustaka

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Metodologi Pengembangan Perangkat Lunak

Bab ini membahas mengenai desain dan perancangan perangkat lunak. Desain perangkat lunak meliputi desain data, arsitektur, proses, dan implementasi dari desain sistem yang dilakukan pada tahap desain.

Bab IV Pengujian dan Evaluasi

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi, dan evaluasi untuk mengetahui kemampuan aplikasi.

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi kode-kode sumber yang penting pada aplikasi ini.

BAB II

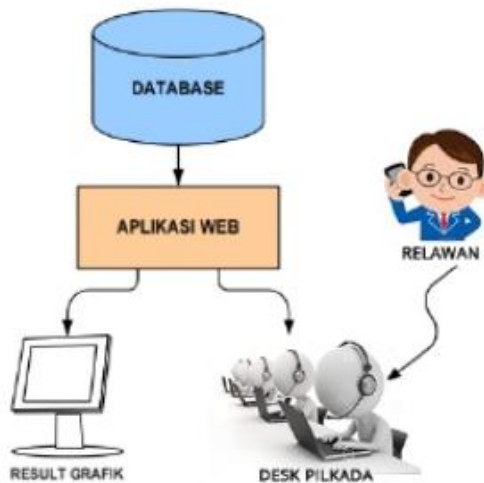
DASAR TEORI

Pada bab ini dibahas mengenai dasar teori yang menjadi dasar pembuatan Tugas Akhir ini. Tinjauan pustaka meliputi HTML5, PhoneGap, Facebook, Facebook Query Language, dan Graph API.

2.1. Penelitian Terkait

Untuk studi penelitian terkait yaitu paper atau penelitian oleh Yoshida Sary, S.Kom, M.Kom yang berjudul “Aplikasi Real Count Pemilihan Kepala Daerah Sebagai Alat Ukur Pemetaan Politik Partai Studi Kasus Tapanuli Utara”. Pada saat pilkada berlangsung dan tahapan pencoblosan surat suara telah selesai maka kemudian tahapan selanjutnya adalah penghitungan surat suara. Relawan setiap TPS akan menunggu sampai proses perhitungan selesai. Kemudian relawan merekap secara manual hasil perhitungan suara didalam satu formulir yang telah disiapkan dari Media center pemenang kandidat. Setelah selesai merekap kemudian relawan mulai melakukan komunikasi secara langsung ke operator aplikasi real count yang telah ditunjuk sesuai dengan kecamatannya.

Selanjutnya data tersebut akan disimpan dalam sebuah database (DB) yang berhubungan langsung dengan sebuah aplikasi web untuk menampilkan secara real-time hasil perhitungan suara yang diperoleh, yang kemudian akan disajikan dalam 2 bentuk, yaitu dalam bentuk grafik dan teks. Aplikasi web ini dikelola oleh seorang administrator, yang bertugas mengontrol kerja aplikasi. Proses kerja aplikasi real count ini dapat digambarkan seperti pada gambar 2.1 berikut ini:[1]



Gambar 2.1 Deskripsi Umum Aplikasi

2.2. Pemilihan Umum

Pemilihan umum (pemilu) di Indonesia pada awalnya ditujukan untuk memilih anggota lembaga perwakilan, yaitu DPR, DPRD Provinsi, dan DPRD Kabupaten/Kota. Setelah amandemen keempat UUD 1945 pada 2002, pemilihan presiden dan wakil presiden (pilpres), yang semula dilakukan oleh MPR, disepakati untuk dilakukan langsung oleh rakyat sehingga pilpres pun dimasukkan ke dalam rangkaian pemilu. Pilpres sebagai bagian dari pemilu diadakan pertama kali pada Pemilu 2004. Pada 2007, berdasarkan Undang-Undang Nomor 22 Tahun 2007, pemilihan kepala daerah dan wakil kepala daerah (pilkada) juga dimasukkan sebagai bagian dari rezim pemilu. Pada umumnya, istilah "pemilu" lebih sering merujuk kepada pemilihan anggota legislatif dan presiden yang diadakan setiap 5 tahun sekali.[2]

2.3. Kepala Daerah

Pengertian Kepala Daerah - Pemerintah Daerah dipimpin oleh Kepala Daerah yang dibantu oleh Wakil Kepala Daerah. Sebutan Kepala Daerah dan Wakil Kepala Daerah adalah

- untuk Daerah Provinsi disebut Gubernur dan Wakil Gubernur;
- untuk Daerah Kabupaten disebut Bupati dan Wakil Bupati;
- untuk Daerah Kota disebut Wali Kota dan Wakil Wali Kota.[3]

Pemilihan kepala daerah (Pilkada atau Pemilukada) dilakukan secara langsung oleh penduduk daerah administratif setempat yang memenuhi syarat. Pemilihan kepala daerah dilakukan satu paket bersama dengan wakil kepala daerah.[4]

2.4. Quick Count

Quick count merupakan proses perhitungan cepat hasil voting menggunakan metode sampling dan kemampuan teknologi komunikasi. Metode sampling sendiri adalah sebuah metode dalam ilmu statistik dan menentukan tingkat ke akurasian hasil dari quick count. Sedangkan yang di maksud dengan teknologi komunikasi yaitu teknologi SMS yang di dukung dengan program SMS-gateway.

Berikut adalah cara kerja quick count dalam proses pemilu:

1. Mempersiapkan sebuah perangkat sistem pendukung untuk memberikan data dengan cepat ke pusat pengolah data lembaga survei yang melakukan metode quick count. Mulai dari komputer untuk menginput data hingga ponsel untuk mengirim SMS hasil pemilu ke server tempat menerima data.
2. Pemilihan TPS sebagai tempat pengambilan data harus proposional. Jika pemilih banyak, maka lokasi sampel (TPS) yang di ambilpun banyak dan mewakili karakteristik populasi.
3. Mempersiapkan relawan untuk mengambil dan menginput sampel ke dalam sistem data. Jumlah relawan

yang di butuhkan pada tahap ini cukup banyak untuk mengambil data dari TPS yang telah di pilih.

4. Data yang di dapatkan di olah ke pusat dengan metode ilmu statistik. Dari olahan data ini lembaga survei dapat menghitung secara cepat siapa pemenang pemilu.[5]

2.5. Metodologi Quick Count

Quick count dilakukan menggunakan teknik *multistage random sampling*. Dimana setiap anggota populasi akan mendapatkan peluang yang sama untuk terpilih sebagai sampel. Model pengacakan dilakukan secara bertingkat yang dilakukan secara porposional berdasarkan besaran populasi jumlah TPS di daerah tersebut sehingga hasil quick count dapat di generalisasi dan mencerminkan hasil pemilu dengan akurasi tinggi.[5]

2.6. Penarikan Sampel Quick Count

Penarikan sampel dalam survei dilakukan dalam dua tingkatan. Pada tingkat pertama sebaran dan jumlah TPS (menggunakan data sebaran TPS pada pemilu yang sedang dilaksanakan) nasional, provinsi, kabupaten atau kota, diurutkan dari yang paling banyak hingga yang paling sedikit jumlah TPSnya. Kemudian di pilih berdasarkan proposisi jumlah TPS. Pada daerah yang memiliki lebih banyak TPS akan diambil sampel TPS yang lebih banyak. Begitupula sebaliknya. Pada tingkat kedua, petugas lapangan akan mendatangi desa atau kelurahan yang dijadikan target akan mendata seluruh TPS di desa atau kelurahan itu. kemudian, petugas akan mengacaknya hingga terpilih satu TPS saja di desa atau kelurahan tersebut.

Hasil perhitungan quick count bukanlah hasil perhitungan dari seluruh TPS, melainkan menggunakan prinsip ilmu statistika.[5]

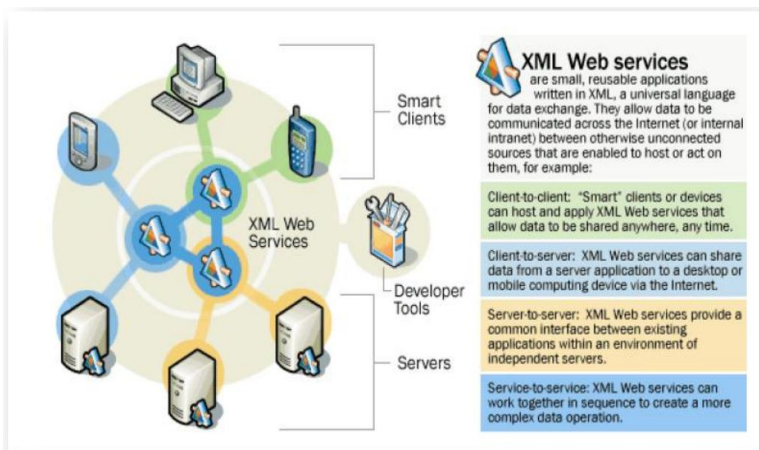
2.7. Web Service

Web service menurut W3.org mendefinisikan web service sebagai “sebuah software aplikasi yang dapat teridentifikasi oleh

URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis XML melalui protokol internet”.

Web service adalah sebuah software aplikasi yang tidak terpengaruh oleh platform, ia akan menyediakan method-method yang dapat diakses oleh network. Ia juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entities bisnis yang berbeda. [6]

Untuk fungsi umum sebuah web service dapat dilihat pada gambar 2.2.[7]



Gambar 2.2 Fungsi dari Web Service

Perbedaan antara web site dengan web service :

WEB SITE	WEB SERVICES
Memiliki web interface	Tidak memiliki interface yang bagus
Dibuat untuk berinteraksi langsung dengan user	Dibuat untuk berinteraksi langsung dengan aplikasi lain
Dibuat untuk bekerja pada	Dibuat untuk bekerja pada semua tipe client aplikasi /

web browser	perangkat device
-------------	------------------

2.8. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi.

Android merupakan generasi baru platform mobile yang memberikan kesempatan pada pengembang untuk melakukan pengembangan program sesuai kebutuhan. Sistem operasi ini merupakan lisensi dibawah naungan GNU, General Public License Versi 2 (GPLv2) atau yang lebih dikenal dengan Copyleft.[8]

Pada Proyek Akhir ini akan digunakan eclipse sebagai IDE untuk pengembangan aplikasi. Sedangkan untuk database program akan digunakan Mysql.

2.9. Eclipse

Eclipse adalah sebuah IDE untuk mengembangkan perangkat lunak dan dijalankan di semua platform. Eclipse sendiri juga termasuk dalam komunitas open source, yang memiliki proyek yang berfokus pada membangun sebuah platform pengembangan terbuka dari extensible framework, tools dan runtime untuk membangun, menyebarkan dan mengelola perangkat lunak di seluruh siklus hidup perangkat lunak tersebut.

Umumnya eclipse digunakan untuk membuat program dalam bahasa pemrograman java, namun eclipse juga bisa digunakan untuk pengembangan program dengan bahasa lainnya.[9]

2.10. Adobe Dreamweaver

Adobe Dreamweaver adalah program yang digunakan untuk membuat atau menyunting halaman web. Software Dreamweaver dikeluarkan oleh Adobe System. Aplikasi ini

banyak digunakan oleh para programmer, desainer dan developer web dikarenakan kemudahan dalam penggunaannya, kelengkapan fiturnya dan juga dukungannya terhadap teknologi terkini.

Adobe Dreamweaver menyediakan fitur editor WYSIWYG (What You See is What You Get) atau dalam bahasa kesehariannya disebut Design View. Maksudnya adalah, tampilan hasil akhir web kita nanti akan sama dengan tampilan pada saat proses perancangan halaman web.

Dengan segala fitur yang ada pada Adobe Dreamweaver, membuat suatu web bukanlah hal yang sulit. Kita tidak perlu menguasai berbagai macam bahasa pemrograman web seperti HTML, CSS, Javascript, PHP, dan sebagainya. Cukup mengetahui dasar dasarnya saja, karena didalam aplikasi ini sudah disediakan alat alat otomatis.[10]

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan membahas tahap analisis kebutuhan bisnis dari sistem yang akan dibangun dan perancangannya. Tahap analisis ini membahas mengenai analisis kebutuhan yang menjadi dasar pada tahap perancangan yang akan dipaparkan pada sub-bab selanjutnya.

3.1. Analisis

Tahap analisis ini terbagi menjadi enam bagian antara lain: domain permasalahan, deskripsi umum perangkat lunak, arsitektur perangkat lunak, spesifikasi kebutuhan perangkat lunak, analisa stakeholder, dan skenario kasus penggunaan perangkat lunak. Berikut penjabaran bagian-bagian tahap analisis.

3.1.1. Domain Permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah bagaimana membuat aplikasi mobile berbasis android yang digunakan oleh petugas di TPS sampel serta membuat aplikasi web yang dipakai untuk menampilkan informasi tentang perolehan suara pasangan calon Pemilu dalam proses quick count atau perhitungan cepat.

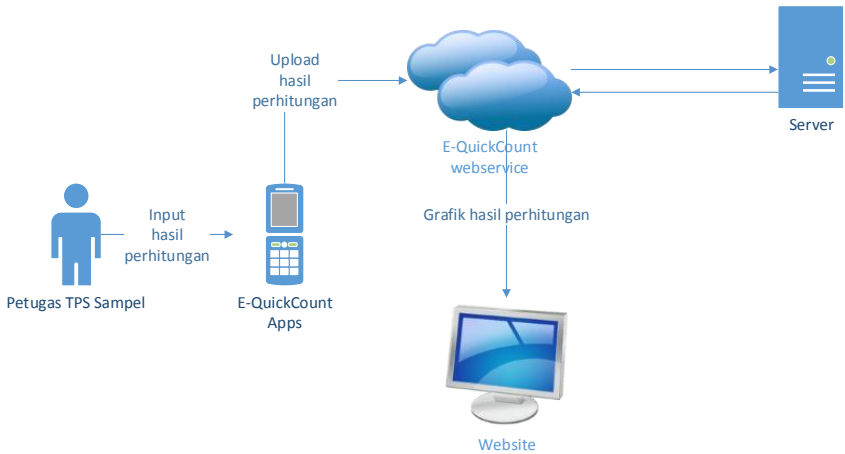
3.1.2. Deskripsi Umum Perangkat Lunak

Pada tugas akhir ini, akan dibangun sebuah aplikasi e-Quickcount yang bisa dijalankan di platform android dan aplikasi ini nantinya akan dapat digunakan secara flexible atau dapat dipakai di semua Pemilukada di Indonesia, baik secara bersamaan antara satu daerah dengan daerah lain maupun tidak.

Keluaran dari Tugas Akhir ini adalah sebuah aplikasi e-Quickcount yang bereksistensi .apk untuk ponsel pintar dengan platform Android, serta aplikasi web untuk menampilkan informasi perolehan suara para calon.

3.1.3. Arsitektur Perangkat Lunak

Ilustrasi arsitektur perangkat lunak pada e-Quickcount dapat dilihat pada Gambar 3.1 merupakan arsitektur sistem, dimana petugas TPS sampel dapat melakukan login pada aplikasi e-QuickCount, setelah itu petugas bisa melakukan entri data hasil perhitungan suara para Calon Pemimpin Kepala Daerah secara portabel atau langsung menggunakan e-QuickCount yang sudah terpasang di ponsel android petugas. Lalu petugas akan mengunggah data tersebut melalui jaringan internet ke e-QuickCount web service. Data tersebut diolah di web service dan selanjutnya hasil akhir pengolahan data akan ditampilkan melalui aplikasi web yang telah disediakan.



Gambar 3.1 Arsitektur Sistem

3.1.4. Spesifikasi Kebutuhan Perangkat Lunak

Pada sistem yang akan dibuat ini, dibutuhkan beberapa fungsi yang dapat membantu proses bisnis dalam sistem. Fungsi-fungsi tersebut antara lain.

1. Login Petugas
Fungsi ini digunakan untuk login petugas di TPS pada aplikasi mobile.
2. Mengunggah Nilai Suara Calon
Fungsi ini digunakan untuk mengunggah nilai suara yang sudah dimasukkan oleh petugas.
3. Mengolah Nilai Suara Calon
Fungsi ini digunakan untuk mengolah nilai suara para calon.
4. Login Admin
Fungsi ini digunakan untuk login admin utama pada aplikasi web.
5. Mendaftarkan Petugas
Fungsi ini digunakan untuk mendaftarkan petugas baru oleh admin.
6. Mengisi Data Pemilu
Fungsi ini digunakan untuk mengisi data-data pemilu yang akan berlangsung.
7. Menampilkan Hasil Perhitungan Cepat
Fungsi ini digunakan untuk menampilkan hasil perhitungan cepat pemilu di aplikasi web.

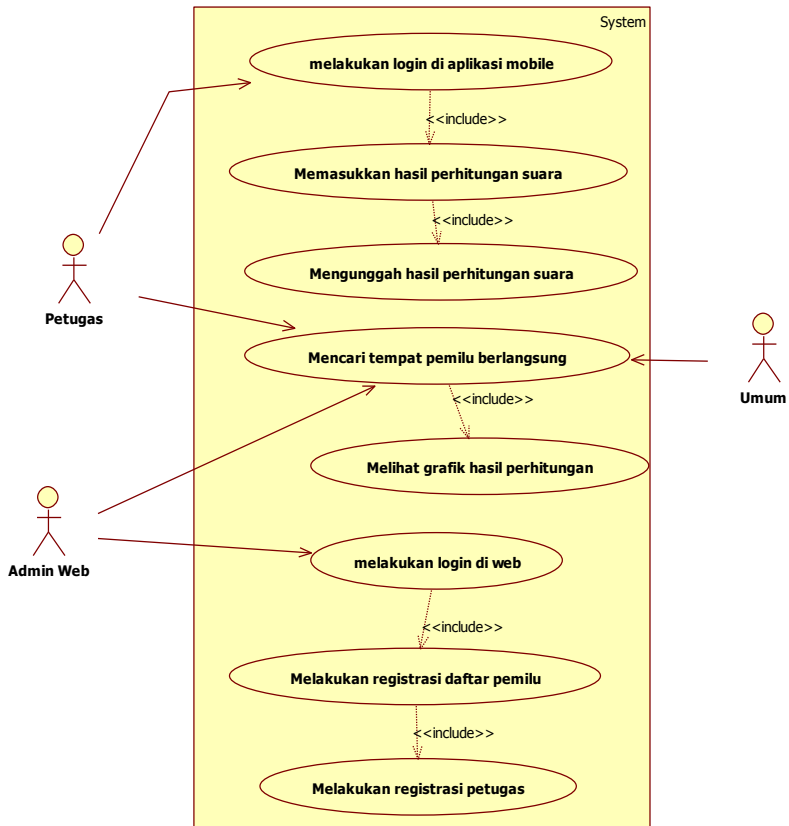
3.1.5. Analisa Stakeholder

Aktor yang terdapat pada aplikasi e-Quickcount merupakan petugas di TPS sampel, yaitu pengguna khusus yang bertugas mengisi data berupa nilai suara calon dan mengunggahnya ke server e-Quickcount. Sedangkan untuk aplikasi webnya merupakan masyarakat umum, dimana masyarakat umum dapat melihat serta memantau langsung jalannya proses perhitungan cepat melalui aplikasi web tersebut.

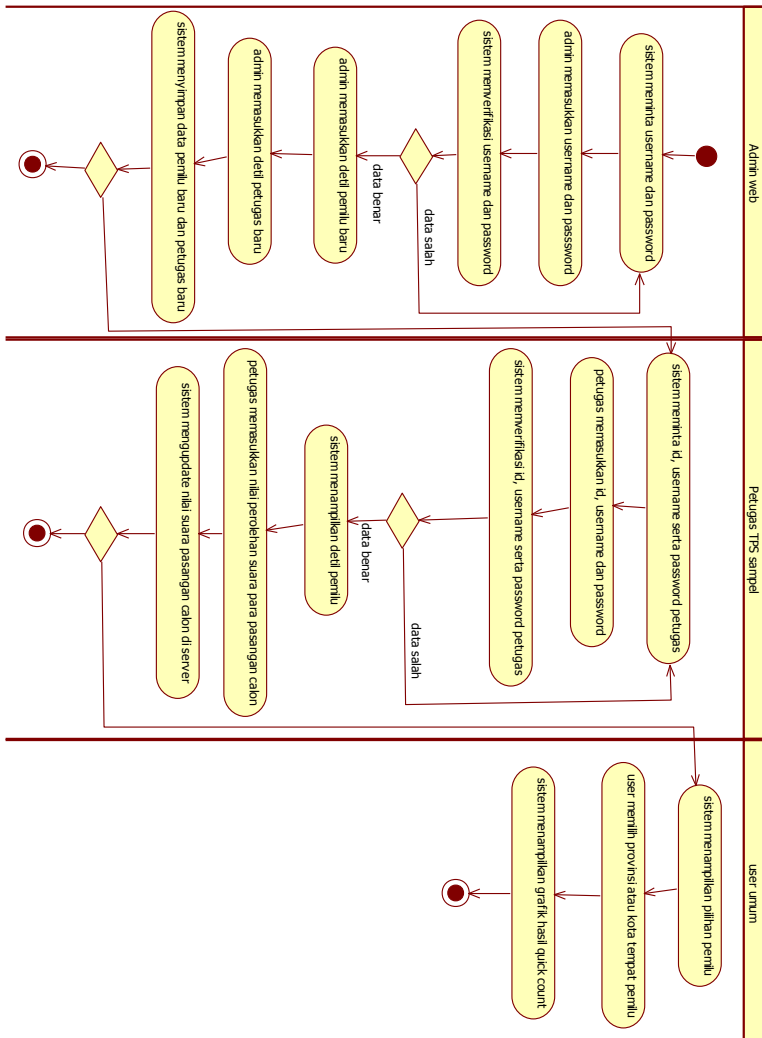
3.1.6. Skenario Kasus Penggunaan

Berdasarkan analisis aktor yang telah dijelaskan, dibuat diagram kasus penggunaan yang menjelaskan fungsionalitas modul mendapatkan data dari basis data pada aplikasi e-

Quickcount yang dikembangkan. Kasus penggunaan dari perangkat lunak ini digambarkan oleh diagram kasus penggunaan yang ada pada Gambar 3.2, dimana ada 3 orang aktor dalam aplikasi ini yaitu petugas TPS sampel, petugas yang memasukkan nilai suara para pasangan calon, admin aplikasi web, dan user umum atau masyarakat umum. Untuk diagram aktivitas menyeluruh dari sistem dapat dilihat pada Gambar 3.3.



Gambar 3.2 Diagram Kasus Penggunaan



Gambar 3.3 Diagram Aktivitas Sistem

Berdasarkan diagram kasus penggunaan pada Gambar 3.2, dapat diketahui bahwa terdapat enam kasus penggunaan, di antaranya adalah Memasukkan hasil perhitungan suara, Mengunggah hasil perhitungan suara, Mencari kota tempat pemilu berlangsung, Melihat grafik hasil perhitungan cepat, Melakukan registrasi petugas, dan Melakukan registrasi daftar pemilu.

Dari Gambar 3.2 dibuat deskripsi kasus penggunaan seperti dilihat pada Tabel 3.1.

Tabel 3.1 Deskripsi Kasus Penggunaan

No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1.	EQC-001	Melakukan login di aplikasi mobile	Petugas TPS lapangan memasukkan ID, username dan password.
2.	EQC -002	Memasukkan hasil perhitungan suara	Petugas lapangan yang ada di TPS memasukkan nilai suara para calon lalu mengunggahnya ke server.
3.	EQC -003	Mencari kota atau provinsi tempat pemilu berlangsung	Pengguna atau masyarakat umum memilih daftar kota-kota maupun provinsi yang ingin diketahui hasil perhitungan cepat pemilu yang sedang berlangsung di kota atau provinsi tersebut.
4.	EQC -004	Melakukan login di aplikasi web	Admin memasukkan username dan password.
5.	EQC -005	Melakukan registrasi pemilu baru	Admin melakukan registrasi atau memasukkan data pemilu yang akan berlangsung.

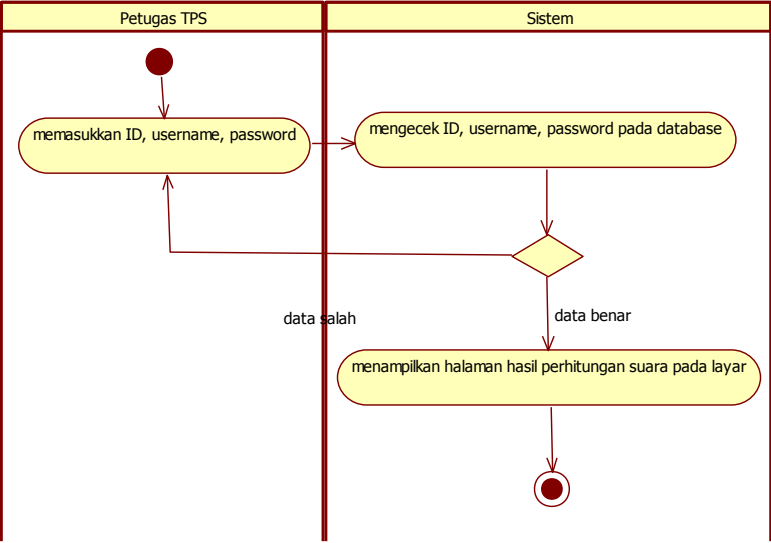
No.	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
6.	EQC -006	Melakukan registrasi petugas TPS sampel	Admin melakukan registrasi atau mendaftarkan petugas yang ada di TPS sampel.

3.1.6.1. Kasus Pengguna Melakukan Login di Aplikasi Mobile

Spesifikasi kasus pengguna melakukan login di aplikasi mobile dijelaskan pada Tabel 3.2.

Tabel 3.2 Skenario Kasus Pengguna: Melakukan Login di Aplikasi Mobile

Nama Kasus Penggunaan	Memasukkan Hasil Perhitungan Suara
Nomor	EQC-001
Deskripsi	Proses melakukan login di aplikasi mobile dilakukan oleh petugas TPS sampel yang sudah terdaftar.
Aktor	Petugas di TPS sampel.
Kondisi Awal	Pengguna sudah terdaftar sebagai petugas TPS sampel.
Alur Normal	<ol style="list-style-type: none"> 1. Sistem meminta pengguna memasukkan ID, username serta password. 2. Pengguna memasukkan data yang diminta system. 3. System mengecek id, username serta password pengguna.
Alur Alternatif	Tidak ada
Kondisi Akhir	Petugas berhasil masuk kedalam aplikasi mobile.



Gambar 3.4 Diagram Aktivitas Melakukan Login Aplikasi Mobile

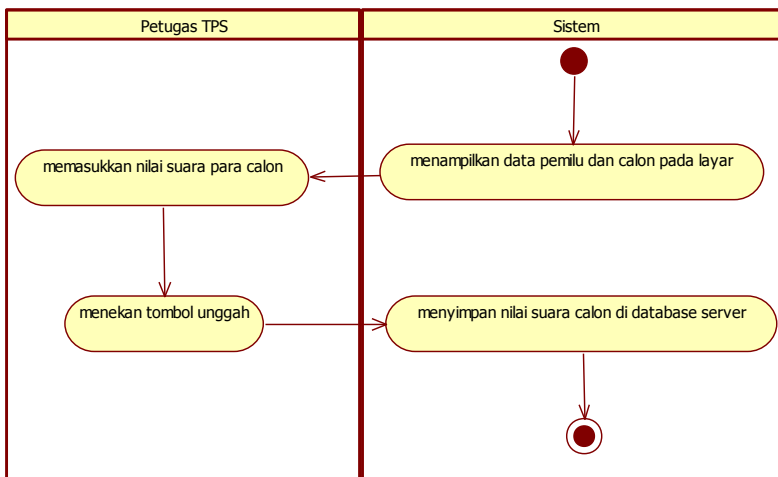
3.1.6.2. Kasus Pengguna Memasukkan Hasil Perhitungan Suara

Spesifikasi kasus pengguna memasukkan hasil perhitungan suara dijelaskan pada Tabel 3.3.

Tabel 3.3 Skenario Kasus Pengguna: Memasukkan Hasil Perhitungan Suara

Nama Kasus Penggunaan	Memasukkan Hasil Perhitungan Suara
Nomor	EQC-002
Deskripsi	Proses memasukkan hasil perhitungan suara ke server melalui jaringan internet oleh petugas TPS sampel.
Aktor	Petugas di TPS sampel.

Nama Kasus Penggunaan	Memasukkan Hasil Perhitungan Suara
Kondisi Awal	Pengguna telah melakukan login di aplikasi mobile.
Alur Normal	<ol style="list-style-type: none"> 1. System menampilkan informasi pemilu serta pasangan calon. 2. System meminta pengguna memasukkan nilai perolehan suara tiap pasangan. 3. Petugas memasukkan nilai suara pasangan calon. 4. Pengguna mengkonfirmasi data untuk kemudian disimpan system di database server.
Alur Alternatif	Tidak ada
Kondisi Akhir	Data yang diunggah disimpan di database server.



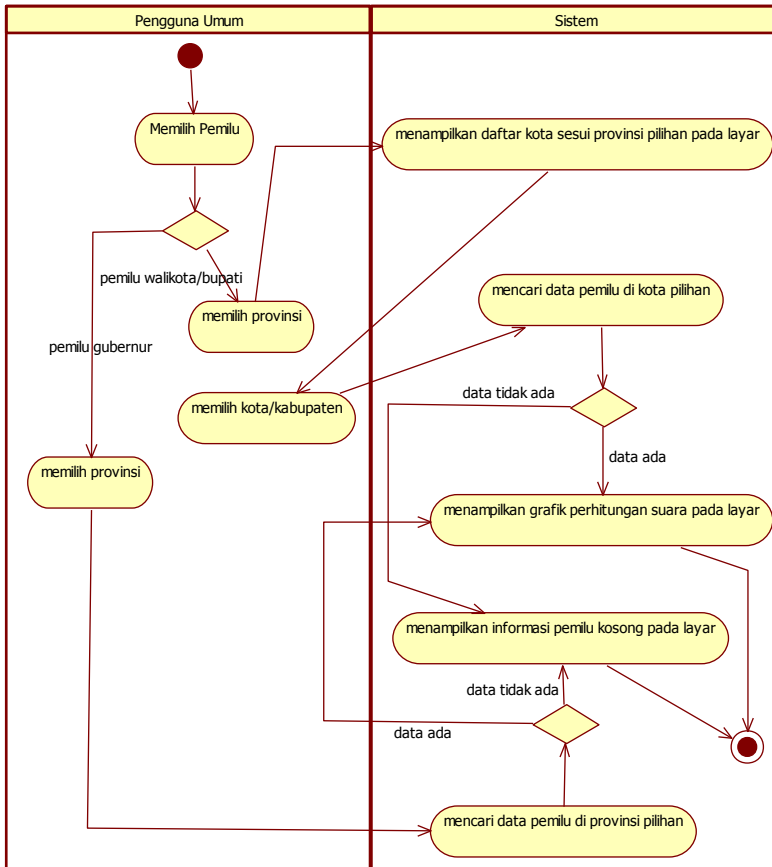
Gambar 3.5 Diagram Aktivitas Memasukkan Hasil Perhitungan Suara

3.1.6.3. Kasus Pengguna Mencari Kota atau Provinsi Tempat Pemilu Berlangsung

Spesifikasi kasus pengguna mencari kota atau provinsi tempat pemilu berlangsung dijelaskan pada Tabel 3.4. Selain mencari kota atau provinsi namun juga melihat hasil perhitungan cepatnya berupa grafik atau tabel.

Tabel 3.4 Skenario Kasus Pengguna: Mencari Kota Atau Provinsi Tempat Pemilu Berlangsung

Nama Kasus Penggunaan	Mencari Kota atau Provinsi Tempat Pemilu Berlangsung
Nomor	EQC-003
Deskripsi	Proses memilih kota atau provinsi yang ada pada list menu sesuai tempat dimana pemilu berlangsung.
Aktor	Masyarakat umum
Kondisi Awal	a. Pengguna telah membuka alamat web e-quickcount pada browser. b. Pengguna berada pada halaman e-quickcount.
Alur Normal	1. Pengguna memilih provinsi tempat pemilu walikota/bupati dilaksanakan. 2. System menampilkan kota/kabupaten sesuai dengan provinsi yang dipilih oleh pengguna. 3. Pengguna memilih kota/kabupaten tempat pemilu yang sedang berlangsung. 4. Pengguna mengkonfirmasi pilihan. 5. System menampilkan data berupa grafik pemilu walikota/provinsi pada layar.
Alur Alternatif	A1. Pengguna memilih provinsi tempat pemilu gubernur dilaksanakan. A1.1 Pengguna mengkonfirmasi pilihan A1.2 System menampilkan data berupa grafik serta table pemilu gubernur pada layar.
Kondisi Akhir	Grafik atau table yang ditampilkan sesuai dengan provinsi maupun kota/kabupaten sesuai pilihan pengguna.



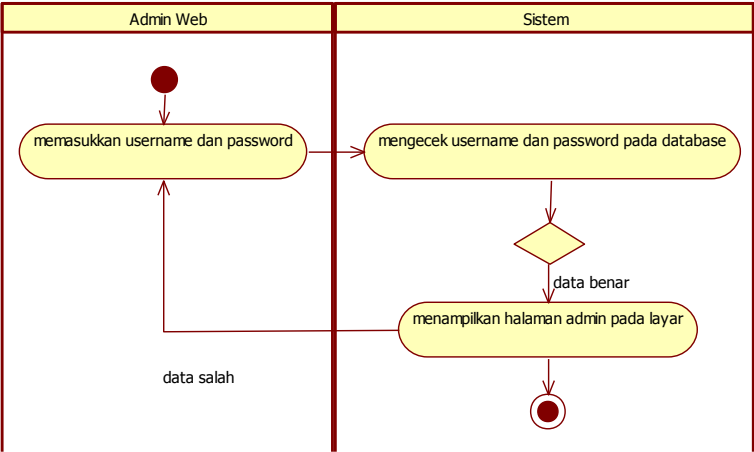
Gambar 3.6 Diagram Aktivitas Mencari Kota Atau Provinsi Tempat Pemilu Berlangsung

3.1.6.4. Kasus Pengguna Melakukan Login di Aplikasi Web

Skenario kasus pengguna melakukan login di aplikasi web dijelaskan pada Tabel 3.5. Login ini dilakukan oleh admin aplikasi web.

Tabel 3.5 Skenario Kasus Pengguna: Melakukan Login Di Aplikasi Web

Nama Kasus Penggunaan	Melakukan Login di Aplikasi web
Nomor	EQC-004
Deskripsi	Proses melakukan login di aplikasi web dilakukan oleh admin web yang sudah terdaftar.
Aktor	Admin aplikasi web.
Kondisi Awal	Pengguna sudah terdaftar sebagai admin aplikasi web.
Alur Normal	<div>1. Sistem meminta pengguna memasukkan username serta password.</div> <div>2. Pengguna memasukkan data yang diminta system.</div> <div>3. System mengecek username serta password pengguna.</div>
Alur Alternatif	Tidak ada
Kondisi Akhir	Petugas berhasil masuk kedalam halaman admin web.



Gambar 3.7 Diagram Aktivitas Melakukan Login Di Aplikasi Web

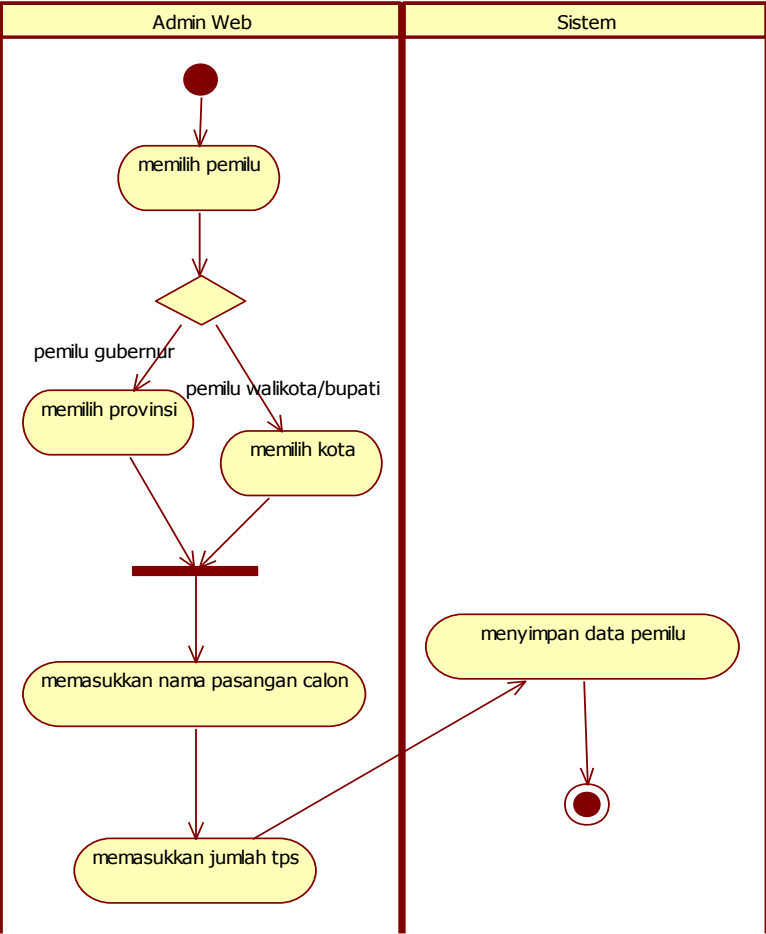
3.1.6.5. Kasus Pengguna Melakukan Registrasi Pemilu Baru

Skenario kasus pengguna melakukan registrasi daftar pemilu dijelaskan pada Tabel 3.6.

Tabel 3.6 Skenario Kasus Pengguna: Melakukan Registrasi Pemilu Baru

Nama Kasus Penggunaan	Melakukan Registrasi pemilu
Nomor	EQC-005
Deskripsi	Proses mendaftarkan pemilu baru yang akan berlangsung.
Aktor	Admin aplikasi web.
Kondisi Awal	Pengguna berada pada halaman admin bagian pendaftaran pemilu baru
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih mendaftarkan pemilu gubernur. 2. System menampilkan daftar provinsi 3. Pengguna memilih provinsi lalu mengkonfirmasi pilihan 4. System meminta pengguna mengisikan nama pasangan calon 5. System meminta pengguna mengisikan jumlah tps 6. Pengguna mengkonfirmasi data untuk kemudian disimpan di database server
Alur Alternatif	<p>A1. Pengguna memilih mendaftarkan pemilu walikota/bupati</p> <p>A1.1 system menampilkan daftar provinsi</p> <p>A1.2 pengguna memilih provinsi lalu mengkonfirmasi pilihan</p> <p>A1.3 system menampilkan daftar kota/kabupaten sesuai provinsi pilihan pengguna</p> <p>A1.4 System meminta pengguna mengisikan nama pasangan calon</p> <p>A1.5 System meminta pengguna mengisikan jumlah tps</p>

Nama Kasus Penggunaan	Melakukan Registrasi pemilu
	A1.6 Pengguna mengkonfirmasi data untuk kemudian disimpan di database server
Kondisi Akhir	Sistem menyimpan daftar pemilu dalam database.



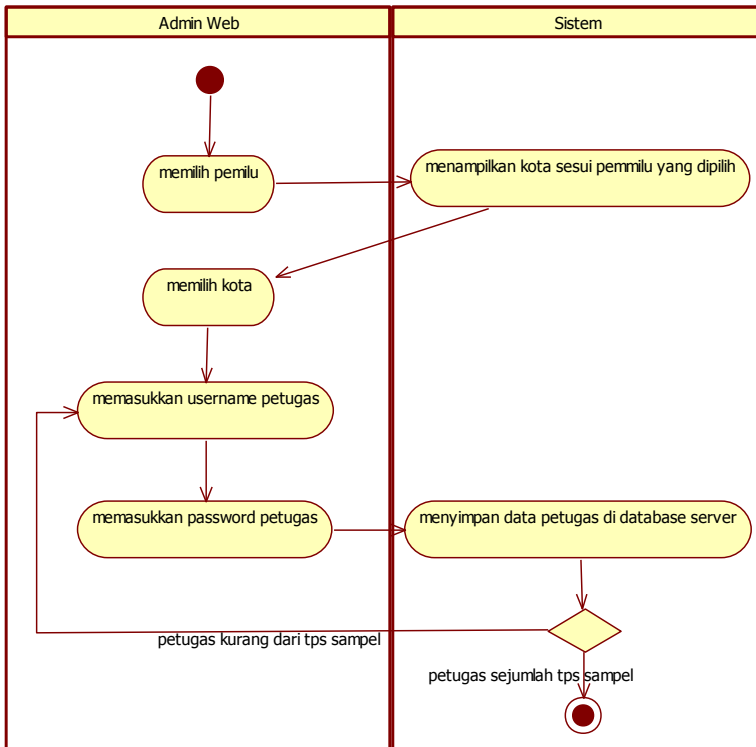
Gambar 3.8 Diagram Aktivitas Melakukan Registrasi Pemilu Baru

3.1.6.6. Kasus Pengguna Melakukan Registrasi Petugas TPS Sampel

Skenario kasus pengguna melakukan registrasi petugas TPS sampel dijelaskan pada Tabel 3.7. Disini aktor atau admin aplikasi web mendaftarkan petugas sesuai jumlah tps dalam pemilu tersebut. Admin memasukkan username serta password petugas.

Tabel 3.7 Skenario Kasus Pengguna: Melakukan Registrasi Petugas TPS Sampel

Nama Kasus Penggunaan	Melakukan Registrasi Petugas TPS Sampel
Nomor	EQC-006
Deskripsi	Proses mendaftarkan petugas TPS sampel kedalam database.
Aktor	Admin aplikasi web.
Kondisi Awal	Pengguna berada pada halaman admin bagian pendaftaran petugas TPS sampel baru
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih daftar pemilu 2. System menampilkan daftar kota sesuai pemilu pilihan pengguna 3. Pengguna memilih kota lalu mengkonfirmasi pilihan 4. System meminta pengguna mengisikan username petugas 5. System meminta pengguna mengisikan password petugas 6. Pengguna menkonfirmasi data untuk disimpan di database server
Alur Alternatif	Tidak ada
Kondisi Akhir	Sistem menyimpan data petugas TPS sampel dalam database.



Gambar 3.9 Diagram Aktivitas Melakukan Registrasi Petugas Tps Sampel

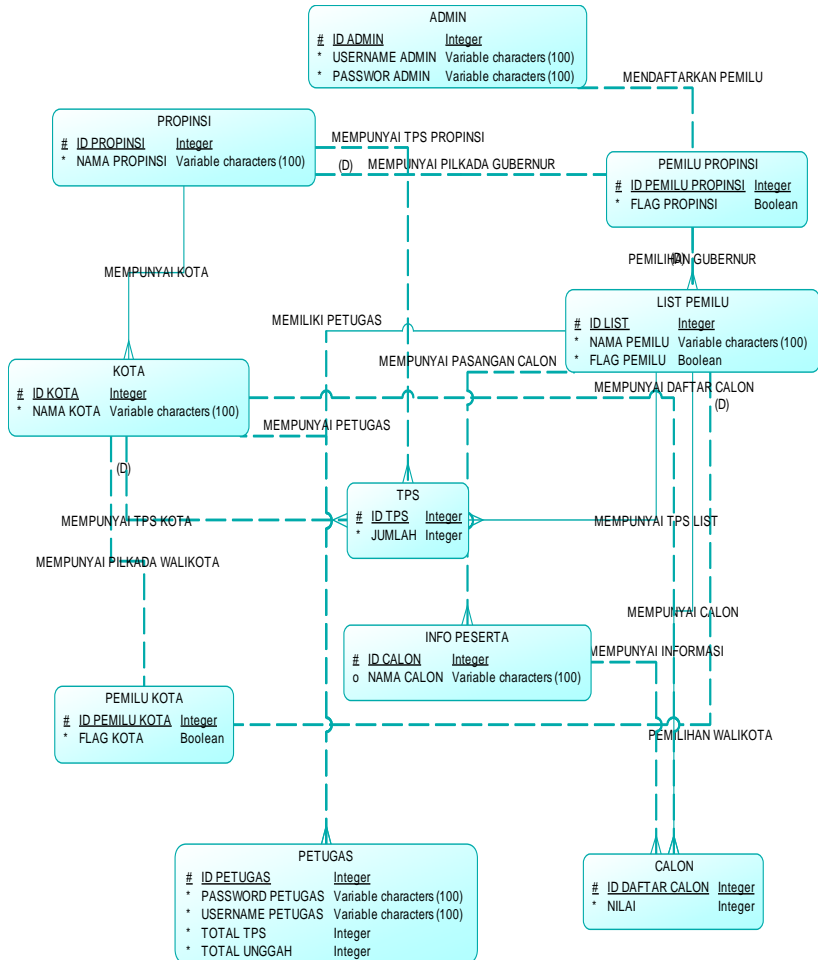
3.2. Tahap Perancangan

Tahap perancangan dalam sub bab ini dibagi menjadi dua bagian yaitu perancangan data, dan perancangan antarmuka grafis.

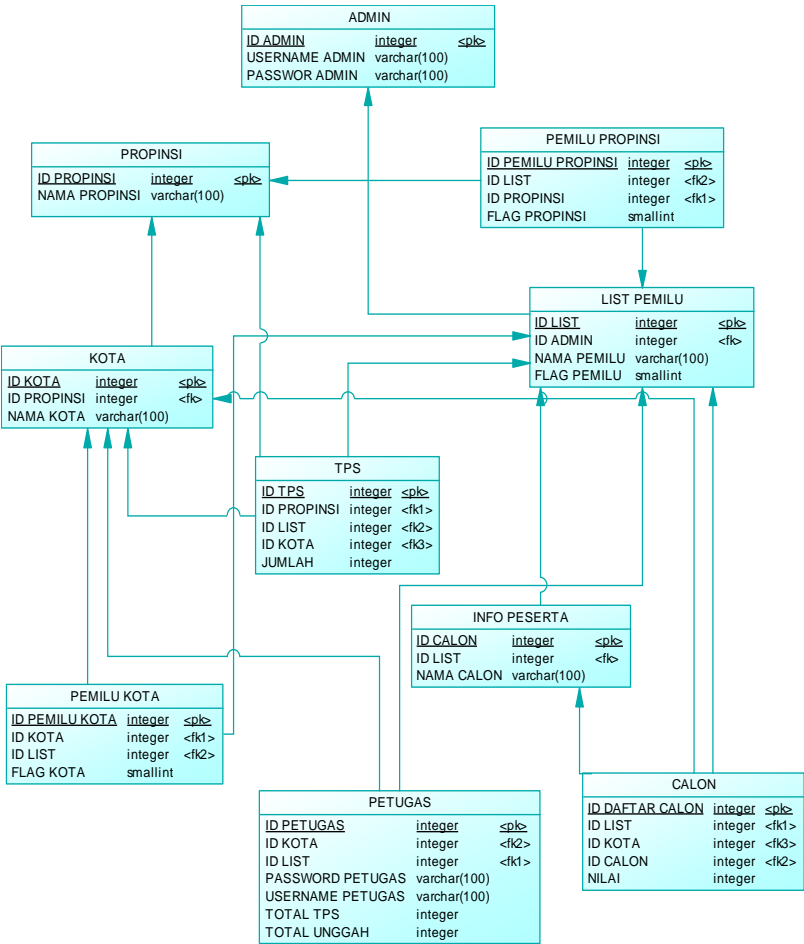
3.2.1. Perancangan Data

Pada subbab ini akan dijelaskan mengenai rancangan data yang akan digunakan untuk pencapaian suatu fungsi pada

program. Subbab ini menampilkan data dalam bentuk *Conceptual Data Model* (CDM) yang digambarkan pada Gambar 3.10 dan *Physical Data Model* (PDM) yang digambarkan pada Gambar 3.11.



Gambar 3.10 CDM



Gambar 3.11 PDM

3.2.3.1 Tabel admin

Tabel ini digunakan untuk menyimpan data admin utama. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.8.

Tabel 3.8 Tabel admin

Nama Kolom	Keterangan
Id_admin	Merupakan <i>primary key</i>
Username	Username admin utama
password	Password yang digunakan

3.2.3.2 Tabel calon

Tabel ini digunakan untuk menyimpan data nilai para calon yang mengikuti pemilu. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.9.

Tabel 3.9 Tabel calon

Nama Kolom	Keterangan
Id_daftarcalon	Merupakan <i>primary key</i>
Id_calon	Merupakan <i>foreign key</i>
Id_list	Merupakan <i>foreign key</i>
Id_kota	Merupakan <i>foreign key</i>
Nilai	Berisi nilai yang diperoleh pasangan hasil dari perhitungan quick count

3.2.3.3 Tabel info_peserta

Tabel ini digunakan untuk menyimpan data para calon yang mengikuti pemilu. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.10.

Tabel 3.10 Tabel info_peserta

Nama Kolom	Keterangan
Id_calon	Merupakan <i>primary key</i>
Nama_calon	Nama pasangan calon
Id_list	Merupakan <i>foreign key</i>

3.2.3.4 Tabel tps

Tabel ini digunakan untuk menyimpan data jumlah serta kota tempat TPS sampel. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.11.

Tabel 3.11 Tabel tps

Nama Kolom	Keterangan
Id tps	Merupakan <i>primary key</i>
Id provinsi	Merupakan <i>foreign key</i>
Id list	Merupakan <i>foreign key</i>
Id kota	Merupakan <i>foreign key</i>
jumlah	Jumlah tps di kota tersebut

3.2.3.5 Tabel kota

Tabel ini digunakan untuk menyimpan data seluruh kota yang ada di Indonesia. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.12.

Tabel 3.12 Tabel kota

Nama Kolom	Keterangan
Id kota	Merupakan <i>primary key</i>
Id provinsi	Merupakan <i>foreign key</i>
Nama_kota	Nama kota

3.2.1.5 Tabel list pemilu

Tabel ini digunakan untuk menyimpan data pemilu yang sedang atau akan dilaksanakan maupun yang sudah terlaksana. Penjelasan mengenai tabel `list_pemilu` dapat dilihat pada Tabel 3.13.

Tabel 3.13 Tabel list_pemilu

Nama Kolom	Keterangan
Id list	Merupakan <i>primary key</i>
Id admin	Merupakan <i>foreign key</i>
Nama_pemilu	Berisi informasi nama pemilu
Flag_pemilu	Sebagai penanda pemilu tersebut sudah atau belum dilaksanakan

3.2.1.6 Tabel pemilu_kota

Tabel ini digunakan untuk menyimpan rekap data pemilu kepala daerah kota atau kabupaten. Penjelasan mengenai tabel `pemilu_kota` dapat dilihat pada Tabel 3.14.

Tabel 3.14 Tabel pemilu_kota

Nama Kolom	Keterangan
Id_pemilu_kota	Merupakan <i>primary key</i>
Id_kota	Merupakan <i>foreign key</i>
Id_list	Merupakan <i>foreign key</i>
Flag_kota	Sebagai penanda pemilu kota atau kabupaten tersebut sudah atau belum dilaksanakan

3.2.1.7 Tabel pemilu_provinsi

Tabel ini digunakan untuk menyimpan rekap data pemilu kepala daerah provinsi. Penjelasan mengenai tabel pemilu provinsi dapat dilihat pada Tabel 3.15.

Tabel 3.15 Tabel pemilu_provinsi

Nama Kolom	Keterangan
Id_pemilu_provinsi	Merupakan <i>primary key</i>
Id_list	Merupakan <i>foreign key</i>
Id_provinsi	Merupakan <i>foreign key</i>
Flag_provinsi	Sebagai penanda pemilu provinsi tersebut sudah atau belum dilaksanakan

3.2.1.8 Tabel petugas

Tabel ini digunakan untuk menyimpan petugas yang ada pada TPS sampel. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.16.

Tabel 3.16 Tabel petugas

Nama Kolom	Keterangan
Id_petugas	Merupakan <i>primary key</i>
Id_list	Merupakan <i>foreign key</i>
Id_kota	Merupakan <i>foreign key</i>
Username	Username petugas
Password	Password yang digunakan
Total_tps	Merupakan total jumlah TPS yang ditangani petugas
Total_unggah	Kali petugas melakukan unggah data

3.2.1.9 Tabel provinsi

Tabel ini digunakan untuk menyimpan data seluruh provinsi yang ada di Indonesia. Penjelasan mengenai tabel tersebut dapat dilihat pada Tabel 3.17.

Tabel 3.17 Tabel provinsi

Nama Kolom	Keterangan
Id_provinsi	Merupakan <i>primary key</i>
Nama_provinsi	Nama provinsi

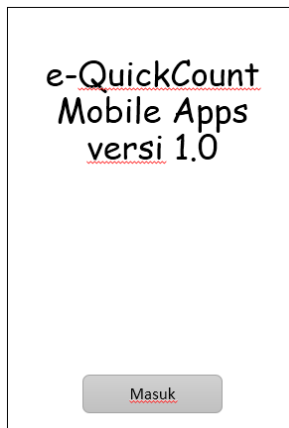
3.2.2. Perancangan Antarmuka Grafis

Pada sub bab ini akan dijelaskan mengenai beberapa rancangan antarmuka utama modul-modul fungsionalitas aplikasi e-QuickCount. Berikut ini penjabaran beberapa rancangan antarmuka tersebut.

3.2.2.1 Rancangan Antarmuka Halaman Pembuka Mobile

Pada halaman ini yang terlihat pada Gambar 3.12, terdapat tulisan e-QuickCount, terdapat juga tombol *masuk* untuk masuk ke dalam aplikasi..

Tujuan adanya antarmuka halaman *pembuka mobile* adalah supaya pengguna bisa mengetahui aplikasi apa yang sedang dibuka.



Gambar 3.12 Rancangan Antarmuka Halaman Pembuka Mobile

3.2.2.2 Rancangan Antarmuka Halaman Login Petugas

Pada halaman ini yang terlihat pada Gambar 3.13, terdapat sebuah tombol *login*, dan tiga *text area* yang harus diisi oleh petugas agar dapat masuk ke halaman unggah nilai, yaitu id petugas, username serta password. Id petugas dipakai untuk menentukan di pemilu apa petugas tersebut ditempatkan.

Tujuan adanya halaman ini adalah untuk login para petugas di TPS sampel.



The image shows a login form with the title "Masukkan Username dan Password" in red text. Below the title, there are three input fields labeled "ID", "Username", and "Password". At the bottom of the form is a grey button labeled "Login".

Gambar 3.13 Rancangan Antarmuka Halaman Login Petugas

3.2.2.3 Rancangan Antarmuka Halaman Unggah Nilai Suara Pasangan Calon

Pada halaman ini terlihat pada gambar 3.14, halaman ini merupakan halaman dimana petugas mengisi nilai suara para calon di *text area* yang telah disediakan. Serta ada sebuah tombol *unggah nilai* yang berfungsi mengunggah nilai yang telah dimasukkan tersebut ke server untuk diolah nantinya. Jumlah *text area* yang ditampilkan sesuai jumlah calon yang ada pada pemilu tersebut.

Tujuan adanya antarmuka ini adalah untuk melakukan fungsionalitas unggah nilai suara para pasangan calon yang dilakukan oleh petugas TPS sampel.

Selamat datang,
PETUGAS

Pasangan 1

Pasangan 2

Pasangan 3

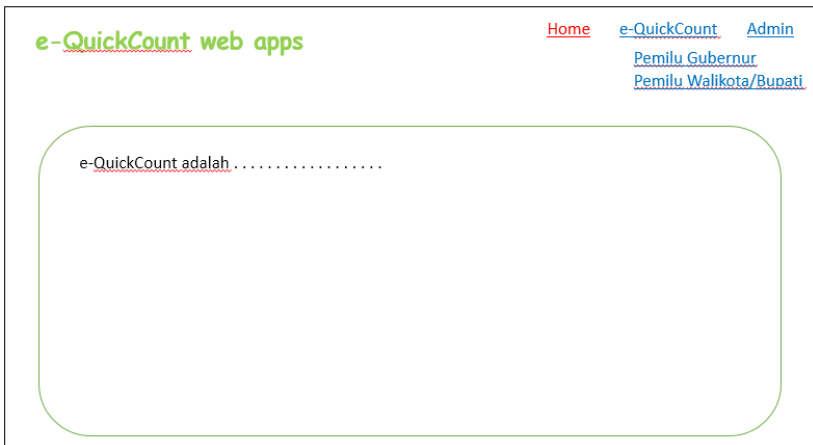
Gambar 3.14 Rancangan Antarmuka Unggah Nilai Suara Pasangan Calon

3.2.2.4 Rancangan Antarmuka Halaman Home Web

Pada halaman ini yang terlihat pada Gambar 3.15 merupakan halaman home dari aplikasi web. Di halaman ini nantinya akan terdapat gambar-gambar yang bertemakan pemilu, semisal logo pemilu maupun *quotes* tentang pemilu. Serta nantinya juga akan ditaruh sebuah artikel tentang pemilu dan informasi tentang e-QuickCount.

Pada pojok sebelah kanan atas akan ada beberapa *link* yang apabila ditekan akan menuju ke halaman *home*, halaman perhitungan cepat pemilu serta menuju ke halaman login untuk admin aplikasi web.

Tujuan adanya antarmuka ini adalah untuk memberitahukan serta menjelaskan tentang pemilu maupun e-QuickCount kepada para pengguna atau masyarakat umum yang membuka aplikasi web.

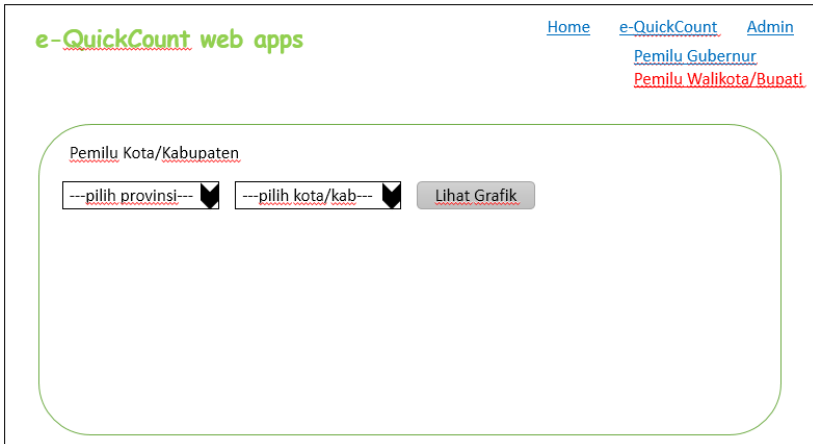


Gambar 3.15 Rancangan Antarmuka Halaman Home Web

3.2.2.5 Rancangan Antarmuka Halaman Pemilu Kota/Kabupaten

Pada konten ini terlihat pada gambar 3.16, terdapat beberapa fitur yang bisa diakses oleh para pengguna, yaitu *dropdownlist* untuk memilih provinsi dan kota/kabupaten tempat dimana Pemilu Kepala Daerah Kota/Kabupaten dilaksanakan. Ada juga sebuah tombol *Lihat Grafik*, dimana tombol ini berfungsi untuk menampilkan hasil grafik perolehan suara para pasangan calon yang mengikuti pemilu kepala daerah kota/kabupaten di daerah yang telah dipilih.

Tujuan adanya antarmuka *Halaman Pemilu Kota/Kabupaten* ini adalah agar para pengguna dapat memilih tempat dimana pemilu kepala daerah kota/kabupaten yang ingin dilihat hasil perhitungan cepatnya.

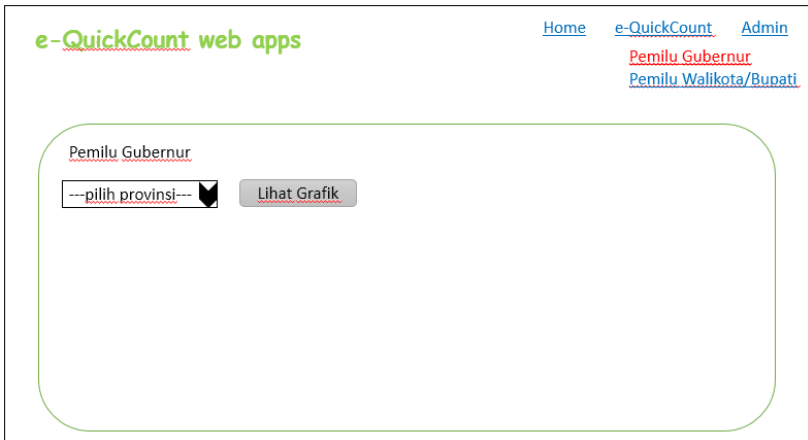


Gambar 3.16 Rancangan Antarmuka Halaman Pemilu Kota/Kabupaten

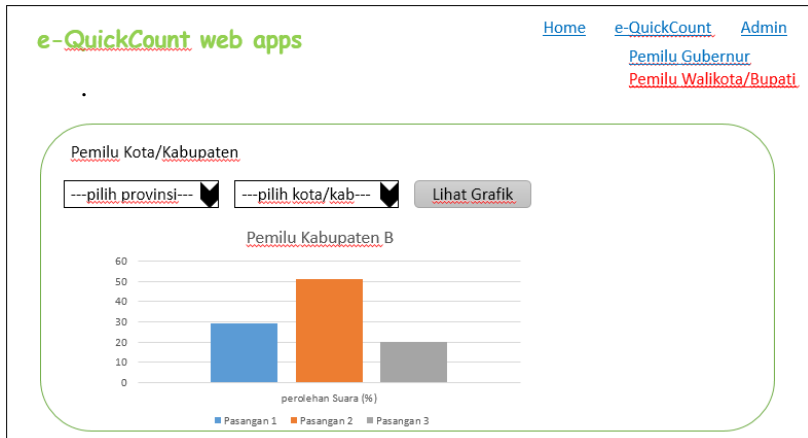
3.2.2.6 Rancangan Antarmuka Halaman Pemilu Provinsi

Pada konten ini terlihat pada gambar 3.17, terdapat beberapa fitur yang bisa diakses oleh para pengguna, yaitu *dropdownlist* untuk memilih provinsi tempat dimana Pemilu Kepala Daerah Provinsi atau Gubernur dilaksanakan. Ada juga sebuah tombol *Lihat Grafik*, dimana tombol ini berfungsi untuk menampilkan hasil grafik perolehan suara para pasangan calon yang mengikuti pemilu kepala daerah provinsi atau gubernur di daerah yang telah dipilih. Pengguna dapat juga memilih kota/kabupaten bila ingin mengetahui perolehan para pasangan calon di suatu kota/kabupaten.

Tujuan adanya antarmuka *Halaman Pemilu Provinsi* ini adalah agar para pengguna dapat memilih tempat dimana pemilu kepala daerah provinsi yang ingin dilihat hasil perhitungan cepatnya.



Gambar 3.17 Rancangan Antarmuka Halaman Pemilu Provinsi



Gambar 3.18 Rancangan Antarmuka Halaman Grafik Pemilu Kota/Kabupaten

3.2.2.7 Rancangan Antarmuka Halaman Grafik Pemilu Kota/Kabupaten

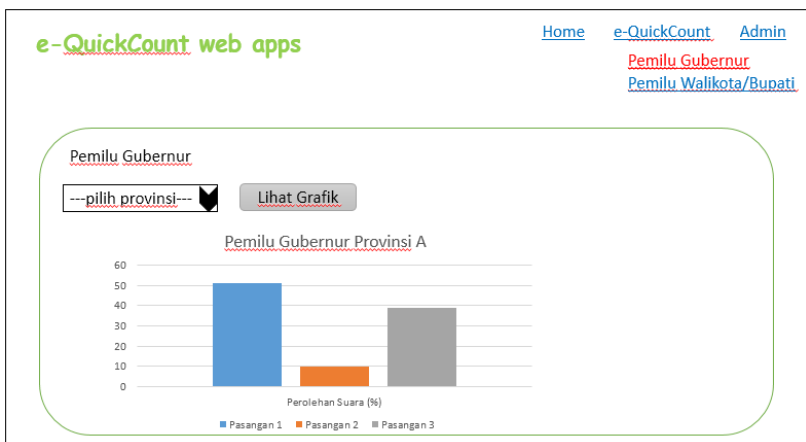
Pada halaman ini yang terlihat pada Gambar 3.18, merupakan halaman yang berisi grafik hasil perhitungan cepat pemilu kepala daerah kota/kabupaten.

Tujuan adanya halaman ini supaya pengguna atau masyarakat umum dapat langsung melihat hasil perhitungan cepat pemilu kepala daerah kota/kabupaten secara langsung.

3.2.2.8 Rancangan Antarmuka Halaman Grafik Pemilu Provinsi

Pada halaman ini yang terlihat pada Gambar 3.19, merupakan halaman yang berisi grafik hasil perhitungan cepat pemilu kepala daerah provinsi.

Tujuan adanya halaman ini supaya pengguna atau masyarakat umum dapat langsung melihat hasil perhitungan cepat pemilu kepala daerah provinsi secara langsung.



Gambar 3.19 Rancangan Antarmuka Halaman Grafik Pemilu Provinsi

3.2.2.9 Rancangan Antarmuka Halaman Login Admin Web

Pada halaman ini yang terlihat pada Gambar 3.20, terdapat *tex area* untuk menuliskan username serta password. Halaman ini dikhususkan untuk admin aplikasi web.

Tujuan adanya halaman ini supaya admin aplikasi web atau admin utama dapat melakukan login.

e-QuickCount web apps

[Home](#) [e-QuickCount](#) [Admin](#)
[Pemilu Gubernur](#)
[Pemilu Walikota/Bupati](#)

Login Admin :

Username

Password

Login

Gambar 3.20 Rancangan Antarmuka Halaman Login Admin Web

e-QuickCount web apps

[Home](#) [e-QuickCount](#) [Admin](#)
[Pemilu Gubernur](#)
[Pemilu Walikota/Bupati](#)

Daftar Pemilu Baru

Jenis Pemilu

Provinsi

Kota

Nama Pasangan 1

Nama Pasangan 2

Nama Pasangan 3

Jumlah TPS sampel

Daftar

Gambar 3.21 Rancangan Antarmuka Halaman Pendaftaran Pemilu

3.2.2.10 Rancangan Antarmuka Halaman Pendaftaran Pemilu

Pada halaman ini yang terlihat pada Gambar 3.21, terdapat beberapa *text area*. Admin memasukkan informasi-informasi tentang pemilu di dalam *text area* yang sudah disediakan.

Tujuan adanya halaman ini adalah supaya admin dapat mendaftarkan pemilu baru yang akan dilaksanakan atau yang direncanakan akan dilaksanakan.

3.2.2.11 Rancangan Antarmuka Halaman Pendaftaran Petugas

Pada halaman ini yang terlihat pada Gambar 3.22, terdapat beberapa *text area*. Admin memasukkan informasi-informasi tentang petugas yang ditempatkan di TPS sampel ke dalam *text area* yang sudah disediakan.

Tujuan adanya halaman ini adalah supaya admin dapat mendaftarkan petugas yang akan diterjunkan ke TPS sampel.

e-QuickCount web apps

[Home](#) [e-QuickCount](#) [Admin](#)
[Pemilu Gubernur](#)
[Pemilu Walikota/Bupati](#)

Daftar Petugas Baru

List Pemilu ▼
Provinsi ▼
Kota ▼
ID
Username
Password

Daftar

Gambar 3.22 Rancangan Antarmuka Halaman Pendaftaran Petugas

3.2.3. Perancangan Webservice

Pada subbab ini akan dijelaskan mengenai beberapa rancangan webservice untuk aplikasi e-QuickCount. Berikut ini penjabaran beberapa rancangan webservice tersebut.

3.2.3.1. Login Aplikasi Mobile

Webservice ini nantinya akan menangani proses login yang dilakukan oleh petugas TPS sampel.

Setelah membuka aplikasi mobile, petugas akan melakukan login dengan memasukkan ID, username serta password. Webservice login aplikasi mobile akan mengecek kebenaran setiap data yang dimasukkan, dengan membandingkannya dengan data yang ada pada server. Jika ada salah satu data yang salah maka webservice akan langsung mengirim peringatan data salah, jika data benar maka webservice akan mengirim format data berupa JSON ke aplikasi mobile yang berisi nama petugas lalu menyimpan data tersebut pada session aplikasi mobile.

3.2.3.2. Menampilkan Keterangan dan Calon Pemilu

Webservice ini nantinya akan menangani proses menampilkan keterangan pemilu berupa nam pemilu serta menampilkan nama para pasangan calon yang maju bersaing.

Setelah melakukan proses login, maka data ID petugas akan dikirim ke webservice menampilkan keterangan dan calon pemilu ini, dimana dalam webservice ada proses mengirim data yang berisi keterangan pemilu dan keterangan pasangan calon ke aplikasi mobile. Data yang di kirim berformat JSON. Data didapat dengan mencari keterangan pemilu yang ada pada database server dengan ID petugas sebagai patokannya. Data ini juga akan disimpan kedalam session aplikasi mobile.

3.2.3.3. Update Perolehan Suara

Webservice ini nantinya akan menangani proses mengupdate nilai suara yang diperoleh oleh masing-masing pasangan calon.

Setelah proses perhitungan suara selesai, maka petugas akan mengisi nilai perolehan suara para pasangan calon dengan cara memasukkan nilai tersebut pada tempat yang sudah disediakan pada aplikasi mobile lalu mengunggahnya ke server. Webservice ini akan mengambil ID calon yang sudah tersimpan pada session aplikasi mobile, lalu mengupdate nilai suara yang sesuai dengan ID calon tersebut dengan cara menjumlahkan dengan nilai sebelumnya.

BAB IV IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi dari perancangan perangkat lunak yang didasari dari Bab 3.

4.1. Lingkungan Implementasi

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Sistem Operasi Windows 7.
2. Android SDK versi 23.0.2.
3. Web Hosting.
4. Starfield phpMyAdmin untuk manajemen basis data.
5. StarUML untuk pembuatan diagram kasus penggunaan.
6. Microsoft Visio untuk pembuatan diagram alir.
7. Power Designer untuk pembuatan CDM dan PDM.
8. Eclipse Java EE Versi Mars untuk pembuatan aplikasi mobile.
9. Dream Weaver CS4 untuk pembuatan web service dan aplikasi web.

4.2. Implementasi Database

Untuk implementasi database dalam sistem ini menggunakan mysql serta PHPMyAdmin sebagai DBMS. Untuk *code* pembuatan tabel adalah sebagai berikut :

```
Table structure for table `admin`--
CREATE TABLE `admin` (
  `id_admin` int(11) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL )
Table structure for table `calon`--
CREATE TABLE `calon` (
  `id_daftarcalon` int(11) NOT NULL,
  `id_calon` int(11) NOT NULL,
```

```

`id_list` int(11) NOT NULL,
`id_kota` int(11) NOT NULL,
`nilai` int(11) NOT NULL,
`foto` mediumblob )
Table structure for table `info peserta`--
CREATE TABLE `info peserta` (
  `id_calon` int(11) NOT NULL,
  `nama_calon` varchar(100) NOT NULL,
  `id_list` int(11) NOT NULL )
Table structure for table `kota`--
CREATE TABLE `kota` (
  `id_kota` int(11) NOT NULL,
  `id_propinsi` int(11) NOT NULL,
  `nama_kota` varchar(100) NOT NULL )
Table structure for table `list pemilu`--
CREATE TABLE `list pemilu` (
  `id_list` int(11) NOT NULL,
  `nama_pemilu` varchar(100) NOT NULL,
  `flag_pemilu` tinyint(1) NOT NULL )
Table structure for table `pemilu kota`--
CREATE TABLE `pemilu kota` (
  `id_pemilu_kota` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `id_list` int(11) NOT NULL,
  `flag_kota` tinyint(1) NOT NULL )
Table structure for table `pemilu propinsi`--
CREATE TABLE `pemilu propinsi` (
  `id_pemilu_propinsi` int(11) NOT NULL,
  `id_list` int(11) NOT NULL,
  `id_propinsi` int(11) NOT NULL,
  `flag_propinsi` tinyint(1) NOT NULL )
Table structure for table `petugas`--
CREATE TABLE `petugas` (
  `id_petugas` int(11) NOT NULL,
  `id_list` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(1024) NOT NULL,
  `flag_petugas` tinyint(1) NOT NULL )
Table structure for table `propinsi`--
CREATE TABLE `propinsi` (
  `id_propinsi` int(11) NOT NULL,
  `nama_propinsi` varchar(100) NOT NULL )
Table structure for table `tps`--
CREATE TABLE `tps` (
  `id_tps` int(11) NOT NULL,
  `id_propinsi` int(11) NOT NULL,
  `id_list` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `jumlah` int(11) NOT NULL )

```

4.3. Implementasi Antar Muka

Implementasi dari antar muka aplikasi ini dibagi menjadi dua, yaitu antar muka untuk aplikasi mobile dan antar muka untuk aplikasi web.

Code untuk antar muka halaman login pada aplikasi mobile adalah sebagai berikut :

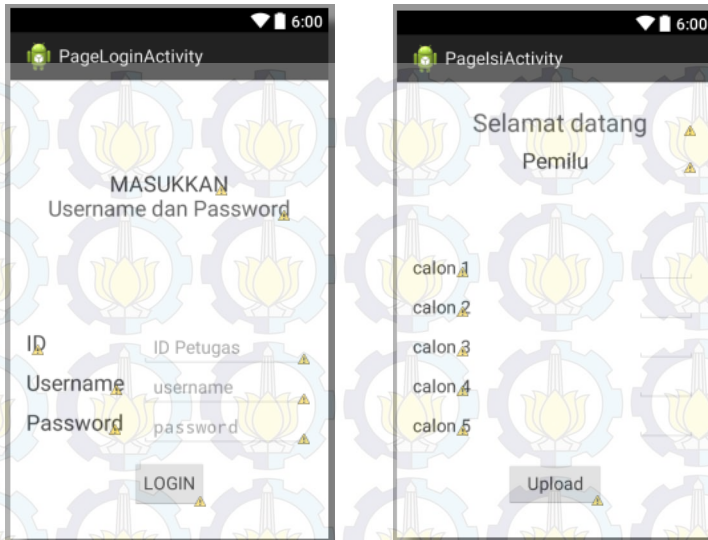
```
<Button
    android:id="@+id/Login"
    android:text="LOGIN" />
<TextView
    android:id="@+id/textView1"
    android:text="MASUKKAN" />
<TextView
    android:id="@+id/textView2"
    android:text="Username dan Password" />
<TextView
    android:id="@+id/textView3"
    android:text="Username" />
<TextView
    android:id="@+id/textView4"
    android:text="Password" />
<EditText
    android:id="@+id/id_petugas"
    android:hint="ID Petugas"
    android:singleLine="true"
    android:inputType="number" />
<EditText
    android:id="@+id/username"
    android:hint="username"
    android:singleLine="true"
    android:inputType="textPersonName" />
<EditText
    android:id="@+id/password"
    android:hint="password"
    android:singleLine="true"
    android:inputType="textPassword" />
<TextView
    android:id="@+id/textView5"
    android:text="ID" />
```

Code untuk antar muka halaman unggah nilai suara calon pada aplikasi mobile adalah sebagai berikut :

```

<TextView
    android:id="@+id/status"
    android:text="Selamat datang" />
<Button
    android:id="@+id/upload"
    android:text="UpLoad" />
<TextView
    android:id="@+id/judul"
    android:text="Pemilu" />
<EditText
    android:id="@+id/nilaicalon1"
    android:inputType="number" />
<TextView
    android:id="@+id/namacalon1"
    android:text="calon 1" />
<TextView
    android:id="@+id/namacalon2"
    android:text="calon 2" />
<TextView
    android:id="@+id/namacalon3"
    android:text="calon 3" />
<TextView
    android:id="@+id/namacalon4"
    android:text="calon 4" />
<TextView
    android:id="@+id/namacalon5"
    android:text="calon 5" />
<EditText
    android:id="@+id/nilaicalon2"
    android:inputType="number" >
<EditText
    android:id="@+id/nilaicalon3"
    android:inputType="number" >
<EditText
    android:id="@+id/nilaicalon4"
    android:inputType="number" >
<EditText
    android:id="@+id/nilaicalon5"
    android:inputType="number" >

```

Gambar 4.1 Antar Muka Halaman Login dan Unggah Nilai

4.4. Implementasi Arsitektur

Implementasi arsitektur sistem berupa arsitektur klien server, dimana webservice dan database yang ditempatkan online pada hosting sebagai server dan para pengguna aplikasi mobile maupun aplikasi web sebagai kliennya.

Pada bagian ini dijelaskan tentang proses pertukaran data antara aplikasi mobile pada ponsel android dengan webservice yang ada pada server. Pertama yaitu pada aplikasi mobile merequest data pemilu untuk ditampilkan pada layar ponsel pintar dengan id, username, dan password petugas TPS sampel sebagai triggernya. Selanjutnya melalui aplikasi mobile mengupdate data nilai perolehan suara yang tersimpan dalam database server. Proses ini dapat dilihat pada Kode Sumber A.1 Lampiran A.

4.5. Implementasi Melakukan Login Aplikasi Mobile

Implementasi melakukan login aplikasi mobile dimulai dari mengambil `id_petugas`, `username` serta `password` dari *text area* yang disediakan. Data tersebut akan diupload ke server untuk dibandingkan dengan data yang ada di *database*. Pembandingannya menggunakan metode AND, dimana ketiga data tersebut harus sama persis dengan yang ada di *database*, jika ada yang tidak sama satu karakterpun akan langsung ditolak. Proses melakukan login aplikasi mobile dapat dilihat pada Kode Sumber A.2 Lampiran A.

```
$id_petugas = $_GET["id_petugas"];
$username   = $_GET["username"];
$password   = $_GET["password"];

$query = "select * from petugas where id_petugas='$id_petugas'
        and username='$username' and password='$password'";
$queryflag = "select * from petugas where id_petugas='$id_petugas'
        and username='$username' and password='$password'";

$hasil = mysql_query($query);
$hasil2 = mysql_query($queryflag);
$flagpetugas = mysql_fetch_array($hasil2);
$flag = $flagpetugas['flag_petugas'];
$idlist = $flagpetugas['id_list'];
if (mysql_num_rows($hasil) > 0 && $flag == 0) {
    $response = array();
    $response["loginhp"] = array();
```

Gambar 4.2 Implementasi Melakukan Login Aplikasi Mobile

4.6. Implementasi Menampilkan Keterangan Pemilu

Implementasi menampilkan keterangan pemilu ini diawali dari mengambil `id_petugas` yang telah melakukan login. Setelah `id_petugas` didapat, maka keterangan pemilu yang berisi nama pemilu serta nama para pasangan calon akan dicari berdasarkan `id_petugas` tersebut. Data berupa

nama_calon, id_calon akan disimpan kedalam *session* menggunakan JSON. Setelah keterangan pemilu didapat, maka akan ditampilkan ke halaman aplikasi mobile. Proses menampilkan keterangan pemilu dapat dilihat pada Kode Sumber A.3 Lampiran A.

```
$query_judul = "select nama_pemilu from 'list pemilu' where id_list='$idlist' ";
$hasiljudul = mysql_query($query_judul);
$judul = mysql_fetch_array($hasiljudul);
$h['nama_pemilu'] = $judul['nama_pemilu'];

$querycalon = "select id_calon from 'info peserta' where id_list='$idlist' ";
$hasilcalon = mysql_query($querycalon);
$calon = mysql_fetch_array($hasilcalon);
$jumlahcalon = mysql_num_rows($hasilcalon);
if ($jumlahcalon == 1)
{
    $querycalonall = "select id_calon from 'info peserta' where id_list='$idlist' ";
    $hasilcalonall = mysql_query($querycalonall);
    $calonall = mysql_fetch_array($hasilcalonall);
    $idcalon = $calonall['id_calon'];
    $h['idcalon1'] = $idcalon;
    $h['idcalon2'] = 0;
    $h['idcalon3'] = 0;
    $h['idcalon4'] = 0;
    $h['idcalon5'] = 0;

    $querycalon1 = "select nama_calon from 'info peserta' where id_list='$idlist' ";
    $hasilcalon1 = mysql_query($querycalon1);
    $calon1 = mysql_fetch_array($hasilcalon1);
    $h['calon1'] = $calon1['nama_calon'];
}
```

Gambar 4.3 Implementasi Menampilkan Keterangan Pemilu

4.7. Implementasi Unggah Data Nilai Suara Pasangan Calon

Implementasi unggah data nilai dimulai dari mengambil nilai yang ada di *text area* calon dan diunggah keserver untuk diolah berdasarkan *id_calon* yang sudah tersimpan sebelumnya didalam *session*. Query yang digunakan berupa update berdasarkan *id_calon*, dan yang diupdate adalah nilai pasangan calon tersebut. Proses unggah data nilai suara pasangan calon dapat dilihat pada Kode Sumber A.4 Lampiran A.

```

$query_ambil_calon5 = "select nilai from calon where id_calon='$idcalon5' and id_kota='$idkota' ";
$hasil5 = mysql_query($query_ambil_calon5);
$data5 = mysql_fetch_array($hasil5);
$Nilai5 = (int)$Nilai5 + (int)$data5['nilai'];

$query = "update calon set nilai = '$Nilai1' where id_calon='$idcalon1' and id_kota='$idkota' ";
$hasil = mysql_query($query);
$query2 = "update calon set nilai = '$Nilai2' where id_calon='$idcalon2' and id_kota='$idkota' ";
$hasil2 = mysql_query($query2);
$query3 = "update calon set nilai = '$Nilai3' where id_calon='$idcalon3' and id_kota='$idkota' ";
$hasil3 = mysql_query($query3);
$query4 = "update calon set nilai = '$Nilai4' where id_calon='$idcalon4' and id_kota='$idkota' ";
$hasil4 = mysql_query($query4);
$query5 = "update calon set nilai = '$Nilai5' where id_calon='$idcalon5' and id_kota='$idkota' ";
$hasil5 = mysql_query($query5);
$query6 = "update petugas set flag_petugas = 1 where id_petugas='$IDP'";
$hasil6 = mysql_query($query6);

```

Gambar 4.4 Implementasi Unggah Nilai

4.8. Implementasi Memilih Tempat Pemilu

Implementasi memilih tempat pemilu dimulai dari memilih nama provinsi atau salah satu provinsi yang telah disediakan pada *dropdownlist* bagian provinsi. Setelah provinsi dipilih maka akan didapat `id_provinsi` dari provinsi yang dipilih tersebut. `Id_provinsi` akan dijadikan acuan untuk menampilkan nama kota/kabupaten di *dropdownlist* bagian kota/kabupaten. Dengan menggunakan query `select` berdasarkan `id_provinsi` maka akan didapat nama kota/kabupaten sesuai dengan provinsi yang dipilih, dan selanjutnya pengguna memilih salah satu dari kota/kabupaten tersebut. Untuk pemilu gubernur pengguna hanya akan memilih provinsi saja, sedangkan untuk pemilu walikota/bupati setelah pengguna memilih provinsi maka pengguna akan diminta untuk memilih kota/kabupatennya juga. Proses memilih tempat pemilu dapat dilihat pada Kode Sumber A.5 pada lampiran A.

```

$quer2="SELECT nama_propinsi,id_propinsi FROM propinsi order by nama_propinsi";
if(isset($cat) and strlen($cat) > 0){
    $quer="SELECT nama_kota FROM kota where id_propinsi=$cat order by nama_kota";
}
else{
    $quer="SELECT nama_kota FROM kota order by nama_kota";
}
echo "<form method=post name=f1 action=''>";
echo "<select name='cat' onchange='reload(this.form)'\><option value=''>Pilih Provinsi</option>";
foreach ($dbo->query($quer2) as $noticia2) {
    if($noticia2['id_propinsi']==@$cat){
        echo "<option selected value='$noticia2[id_propinsi]'\>$noticia2[nama_propinsi]</option>". "<BR>";
        $prop=$_POST['cat'];
    }
    else {
        echo " <option value='$noticia2[id_propinsi]'\>$noticia2[nama_propinsi]</option>";
        $prop=$_POST['cat'];
    }
}
echo "</select>"; echo " ";
echo "<select name='subcat'\><option value=''>Pilih Kota/Kabupaten</option>";
foreach ($dbo->query($quer) as $noticia) {
    echo " <option value='$noticia[nama_kota]'\>$noticia[nama_kota]</option>";
    $kot=$_POST['subcat'];
}
echo "</select>"; echo " ";

```

Gambar 4.5 Implementasi Memilih Tempat Pemilu

4.9. Implementasi Menampilkan Grafik Pemilu Kota/Kabupaten

Implementasi menampilkan grafik pemilu kota/kabupaten menggunakan *jquery highchart*. Proses dimulai dengan mengambil *id_kota* dari proses mengolah *dropdownlist* bagian kota. Setelah *id_kota* didapat maka akan dicari di dalam tabel pemilu untuk melakukan *checking* apakah ada pemilu walikota/bupati di kota tersebut, jika ada maka akan diambil *id_list* dari tabel pemilu yang nantinya akan dilanjutkan untuk mengecek nilai dan nama calon berdasarkan *id_list* tersebut. Nilai dan nama calon yang di dapat akan ditampilkan menggunakan *jquery highchart* kedalam halaman web. Proses menampilkan grafik pemilu kota/kabupaten dapat dilihat pada Kode Sumber A.6 Lampiran A.

```

$sqlcekpemilukota = mysql_query("SELECT * FROM 'pemilu kota' where id_kota='$idkota'") or die (mysql_error());
$dapemilukota = mysql_fetch_array($sqlcekpemilukota);
$flagkota = $dapemilukota['flag_kota'];
$adapemilukota = mysql_num_rows($sqlcekpemilukota);
if ($adapemilukota > 0 && $flagkota == 1) {
    $idpemilukota = $dapemilukota['id_list'];
    $sql = mysql_query("SELECT * FROM 'info peserta' where id_list='$idpemilukota'") or die (mysql_error());
    $sqltotal = mysql_query("SELECT SUM(nilai) AS total_suara from calon where id_list='$idpemilukota'")
    or die (mysql_error());
    $totalnilai = mysql_fetch_array($sqltotal);
    $nilai = (float)$totalnilai['total_suara'];
    while ($data = mysql_fetch_array($sql)) {
        $namacalon = $data['nama_calon'];
        $idcalon = $data['id_calon'];
        $sqljumlahsuara = mysql_query("SELECT nilai FROM calon WHERE id_calon='$idcalon'") or die (mysql_error());
        while ($datajumlah = mysql_fetch_array($sqljumlahsuara)) {
            $jumlah = $datajumlah['nilai'];
        }
    }
    {
        name: '<?php echo $namacalon; ?>',
        data: [<?php
            echo number_format($jumlah/$nilai*100,2);
            ?>]
    },
}

```

Gambar 4.6 Implementasi Menampilkan Grafik Pemilu Kota/Kabupaten

4.10. Implementasi Menampilkan Grafik Pemilu provinsi

Implementasi menampilkan grafik pemilu provinsi menggunakan *jquery highchart*. Proses dimulai dengan mengambil `id_provinsi` dari proses mengolah *dropdownlist* bagian provinsi. Setelah `id_provinsi` didapat maka akan dicari di dalam tabel pemilu untuk melakukan *checking* apakah ada pemilu gubernur di provinsi tersebut, jika ada maka akan diambil `id_list` dari tabel pemilu yang nantinya akan dilanjutkan untuk mengecek nilai dan nama calon berdasarkan `id_list` tersebut. Nilai dan nama calon yang di dapat akan ditampilkan menggunakan *jquery highchart* kedalam halaman web. Proses menampilkan grafik pemilu provinsi dapat dilihat pada Kode Sumber A.7 Lampiran A.


```

$sqlcekpemiluprov2 = mysql_query("SELECT * FROM `pemilu propinsi` where id_propinsi='$searchprovinsi'")
or die (mysql_error());
$datapemiluprov2 = mysql_fetch_array($sqlcekpemiluprov2);
$flagprov2 = $datapemiluprov2['flag_propinsi'];
$adapemiluprov2 = mysql_num_rows($sqlcekpemiluprov2);
if ($adapemiluprov2 > 0 && $flagprov2 == 1) {
    $idpemiluprov2 = $datapemiluprov2['id_list'];
    $sqlnama2 = mysql_query("SELECT * FROM `info peserta` where id_list='$idpemiluprov2'" or die (mysql_error()));
    $sqltotal2 = mysql_query("SELECT SUM(nilai) AS total_suara2 from calon where id_list='$idpemiluprov2'")
    or die (mysql_error());
    $totalnilai2 = mysql_fetch_array($sqltotal2);
    $nilai2 = (float)$totalnilai2['total_suara2'];
    while ($data2 = mysql_fetch_array($sqlnama2)) {
        $namacalon2 = $data2['nama_calon'];
        $idcalon2 = $data2['id_calon'];
        $sqljumlahsuara2 = mysql_query("SELECT SUM(nilai) AS total_suara FROM calon WHERE id_calon='$idcalon2'
        and id_list='$idpemiluprov2'" or die (mysql_error()));
        while ($datajumlah2 = mysql_fetch_array($sqljumlahsuara2)) {
            $jumlah2 = $datajumlah2['total_suara'];
        }
    }
    {
        name: '<?php echo $namacalon2; ?>',
        data: [<?php
            echo number_format($jumlah2/$nilai2*100,2);

```

Gambar 4.7 Implementasi Menampilkan Grafik Pemilu Provinsi

4.11. Implementasi Menghitung Prosentase Nilai Suara yang Masuk

Implementasi dari proses ini dimulai dari mengambil `id_list` dari pemilu yang telah dipilih dalam proses sebelumnya yaitu proses menampilkan grafik. Setelah `id_list` didapat maka akan dilanjutkan dengan mengambil total jumlah petugas yang ada dalam pemilu yang dipilih tersebut dari tabel `tps` berdasarkan `id_list`. Prosentase suara yang sudah masuk didapat dari jumlah petugas yang sudah melakukan pengunggahan data nilai yang ditandai dengan `flag_petugas` bernilai 1 dibagi jumlah total petugas yang ada dalam pemilu tersebut dan dikalikan 100%. Proses menghitung prosentase nilai suara yang masuk dapat dilihat pada Kode Sumber A.8 pada lampiran A.

```

$idpemukota = $dataflagkota['id_list'];
$sqlcektotpet = mysql_query("SELECT * FROM petugas where id_list='$idpemukota'") or die (mysql_error());
$totalpetugas = mysql_num_rows($sqlcektotpet);
$sqlcektotpetupload = mysql_query("SELECT * FROM petugas where id_list='$idpemukota' and flag_petugas=1 ")
or die (mysql_error());
$totalpetugasupload = mysql_num_rows($sqlcektotpetupload);
$prosentase = number_format($totalpetugasupload/$totalpetugas*100,2);
echo "Data masuk ".$prosentase."% suara.";

```

Gambar 4.8 Implementasi Menghitung Prosentase Nilai Suara Masuk

4.12. Implementasi Login Admin Aplikasi Web

Implementasi melakukan login admin aplikasi web dimulai dari mengambil username serta password dari *text area* yang disediakan. Data tersebut akan diupload ke server untuk dibandingkan dengan data yang ada di databse. Pembandingannya menggunakan metode AND, dimana kedua data tersebut harus sama persis dengan yang ada di database, jika ada yang tidak sama satu karakterpun akan langsung ditolak. Proses melakukan login admin aplikasi web dapat dilihat pada Kode Sumber A.9 Lampiran A.

```

$username = stripslashes($username);
$password = stripslashes($password);
$username = mysql_real_escape_string($username);
$password = mysql_real_escape_string($password);

$db = mysql_select_db("quickcount", $connection);

$query = mysql_query("select * from admin where password='$password' AND username='$username'", $connection);
$rows = mysql_num_rows($query);
if ($rows == 1) {
    $_SESSION['login_user']=$username;
    header("location: admin.php");
} else {
    $error = "Username atau Password belum terdaftar";
}
mysql_close($connection);
}

```

Gambar 4.9 Implementasi Login Admin Aplikasi Web

4.13. Implementasi Mendaftar Pemilu Baru

Implementasi mendaftar pemilu baru menggunakan query insert, dimana data pemilu yang akan dimasukkan atau didaftarkan didapat dari *text area* yang ada atau disediakan dalam halaman mendaftar pemilu baru. Data atau informasi yang didaftar berupa jenis pemilu, tempat pemilu dilangsungkan, nama para pasangan calon, serta jumlah petugas yang nantinya akan diterjunkan. Proses mendaftar pemilu baru dijelaskan pada Kode Sumber A.10 Lampiran A.

```
$namapemilu = "Pemilihan Umum Kepala Daerah ".$namakota;
$sqlinsert_listpemilu = mysql_query("insert into 'list pemilu' (nama_pemilu, flag_pemilu) values ('$namapemilu', 0)");
$insert1=1;

$sqlgetidlist = mysql_query("SELECT id_list FROM 'list pemilu' where nama_pemilu='$namapemilu' and flag_pemilu=0 ") or die (mysql_error());
$dataidlist = mysql_fetch_array($sqlgetidlist);
$idlist = $dataidlist['id_list'];

$sqlinsert_pemilukota = mysql_query("insert into 'pemilu kota' (id_kota, id_list, flag_kota) values ('$idkota', '$idlist', 0)");
$insert2=1;

$namacalon1 = $_POST['calon1'];
$namacalon2 = $_POST['calon2'];
$namacalon3 = $_POST['calon3'];
$namacalon4 = $_POST['calon4'];
$namacalon5 = $_POST['calon5'];

if (strlen($namacalon1) > 1)
{
    $sqlinsert_infopeserta = mysql_query("insert into 'info peserta' (nama_calon, id_list) values ('$namacalon1', '$idlist')");
    $sqlgetidcalon1 = mysql_query("SELECT id_calon FROM 'info peserta' where nama_calon='$namacalon1' and id_list='$idlist' ")
    or die (mysql_error());
    $dataidcalon1 = mysql_fetch_array($sqlgetidcalon1);
    $idcalon1 = $dataidcalon1['id_calon'];
    $sqlinsert_calon1 = mysql_query("insert into calon (id_calon, id_list, id_kota, nilai) values ('$idcalon1', '$idlist', '$idkota', 0)");
}
```

Gambar 4.10 Implementasi Mendaftar Pemilu Baru

4.14. Implementasi Mendaftar Petugas

Implementasi mendaftar petugas menggunakan query insert, dimana data yang dimasukkan berupa username, password serta id_list. Data didapat dari *text area* yang telah disediakan. Proses mendaftar petugas dijelaskan pada Kode Sumber A.11 Lampiran A.

```

if (mysql_num_rows($sqlcekjumlah) == $jumlahtps)
{
    ?>
    <script language="JavaScript">
        alert('Semua petugas di kota tersebut telah diinput!');
        document.location='adminpetugasbaru.php';
    </script>
</php>
}

else
{
    $sqlinsert_petugasbaru = mysql_query("insert into petugas (id_list, id_kota, username, password, flag_petugas)
        values ('$idlist', '$idkota', '$username', '$password', 0)") or die (mysql_error());
    $insert1 = 1;
    if ($insert1 == 1) // && $insert2 == 1 && $insert3 == 1 && $insert4 == 1)
    { ?>
        <script language="JavaScript">
            alert('Data TPS Pemilu Provinsi Berhasil diinput!');
            document.location='adminpetugasbaru.php';
        </script>
        </php>
    }
}

```

Gambar 4.11 Implementasi Mendaftar Petugas

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas tentang rangkaian pengujian dan evaluasi perangkat lunak yang dilakukan dari hasil implementasi. Pengujian dilakukan untuk menguji secara keseluruhan apakah semua fungsionalitas berjalan sesuai keinginan. Perangkat lunak dipengujian dengan menggunakan beberapa macam skenario. Pembahasan pada bab ini meliputi lingkungan pengujian, dasar pengujian, skenario pengujian, hasil pengujian, dan evaluasi.

5.1. Lingkungan Pelaksanaan Pengujian

Lingkungan pengujian merupakan perangkat tempat pengujian sistem dilakukan. Lingkungan pengujian ini menggunakan 1 unit ponsel pintar Android untuk pengujian aplikasi mobile dan 1 unit laptop untuk pengujian aplikasi web. Spesifikasi lingkungan pengujian terbagi menjadi 2, yaitu lingkungan perangkat keras dan lingkungan perangkat lunak.

5.1.1. Lingkungan Perangkat Ponsel Pintar

Perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini tidak membutuhkan spesifikasi yang tinggi. Perbedaan yang terlihat saat menggunakan spesifikasi yang berbeda hanya pada tingkat kecepatan prosesnya saja. Adapun spesifikasi perangkat keras yang digunakan pada pengujian ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Lingkungan Perangkat Keras

No	Jenis Perangkat	Perangkat Keras	Spesifikasi
1	Ponsel Android	Model	Samsung ACE 2
		Prosesor	NovaThor U8500 Dual-core 800 MHz
		RAM	768 MB
2	Laptop	Model	Compaq CQ35

		prosesor	Intel Core2Duo T6600 2.20 GHz
		RAM	2 GB

5.1.2. Lingkungan Perangkat Lunak

Aplikasi ini dibangun dengan menggunakan Eclipse Java EE IDE, Versi Mars untuk aplikasi mobile. Dan Dreamweaver CS4 untuk aplikasi web.

Adapun spesifikasi perangkat lunak yang digunakan pada pengujian ini dapat dilihat pada Tabel 5.2

Tabel 5.2 Lingkungan Perangkat Lunak

No	Jenis Perangkat	Perangkat Lunak	Spesifikasi
1	Ponsel Android	Sistem Operasi	Android Jelly Bean 4.1
		Akses Internet	Kecepatan 1 Mbps
		Provider	Telkom Speedy
2	Laptop	Sistem Operasi	Windows 7
		Akses Internet	Kecepatan 1 Mbps
		Provider	Telkom Speedy

5.2. Dasar Pengujian

Pengujian pada e-QuickCount ini dilakukan dengan menggunakan sebuah ponsel dan komputer. Pengujian perangkat lunak ini menggunakan metode pengujian *black box* yang berfokus pada kebutuhan fungsional dan non-fungsional perangkat lunak tersebut. Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasi dan bekerja seperti yang semestinya.

Dilakukan juga pengujian secara subjektif untuk mendapatkan respon pengguna terhadap fungsionalitas dan antarmuka e-QuickCount. Pengujian dilakukan dengan memberikan survei kepada sepuluh orang pengguna. Kemudian

dicatat respon pengguna apakah sudah memenuhi tujuan Tugas Akhir atau belum.

5.3. Skenario Pengujian

Pada subbab ini dijelaskan mengenai skenario pengujian yang digunakan. Seperti yang telah dijelaskan sebelumnya bahwa pengujian yang dilakukan adalah pengujian fungsionalitas untuk menguji kebutuhan fungsional sistem.

5.3.1 Pengujian Fungsional

5.3.1.1. Pengujian Melakukan Login Aplikasi Mobile

Pada bagian ini dijelaskan proses pengujian untuk melakukan login pada aplikasi mobile. Tabel 5.3 menyajikan prosedur pengujian proses melakukan login. Pengujian dilakukan untuk menguji apakah pengguna yang melakukan proses login akan berhasil masuk ke dalam e-QuickCount.

Tabel 5.3 Pengujian Melakukan Login Aplikasi Mobile

Test ID		EQ-MW-01		
Tujuan Test		Mengecek apakah fungsi login pada aplikasi mobile berjalan normal		
Kondisi Awal		Pengguna mengakses halaman login e-QuickCount		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data Pengguna berupa ID petugas 70001, username erlangga serta password erlangga.	Pengguna memasukkan ID, username serta password pada kolom yang telah disediakan.	Data yang telah dimasukkan oleh pengguna dicek ke dalam database dan pengguna masuk ke halaman utama <i>e-QuickCount</i> .	Pengguna berhasil masuk kehalaman utama <i>e-QuickCount</i> .	Proses melakukan login berhasil.



Gambar 5.1 Pengujian Login Aplikasi Mobile

5.3.1.2. Pengujian Mengunggah Nilai Suara Calon

Pada bagian ini akan dijelaskan proses pengujian untuk mengunggah nilai suara calon. Tabel 5.4 menyajikan prosedur pengujian proses mengunggah nilai suara calon. Pengujian dilakukan untuk menguji apakah jika pengguna melakukan unggah nilai suara calon, maka nilai tersebut bisa masuk ke dalam database.

Tabel 5.4 Pengujian Mengunggah Nilai Suara Calon

Test ID		EQ-MW -02		
Tujuan Test		Melakukan pengujian untuk menguji apakah jika nilai suara calon yang diunggah oleh pengguna berhasil masuk ke dalam database.		
Kondisi Awal		Pengguna telah masuk ke halaman utama e-QuickCount		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data berupa nilai suara para pasangan	Pengguna memasukkan nilai atau angka pada	Data berupa angka tersebut akan dijumlahkan	Data nilai suara tersimpan ke dalam	Proses mengunggah nilai suara calon

calon, rasiyo & lucy 50 dam risma wisnu 300	kolom yang telah disediakan	atau diupdate kedalam perolehan suara pasangan calon di database.	database	berhasil.
---	-----------------------------------	---	----------	-----------

Unggah Nilai Suara

Selamat datang
erlangga

Pemilihan Umum Kepala
Daerah Kota Surabaya

Rasiyo & Lucy 50

Risma & Wisnu 300

Upload

Gambar 5.2 Pengujian Mengunggah Nilai

5.3.1.3. Pengujian Melakukan Login Admin Aplikasi Web

Pada bagian ini dijelaskan tentang pengujian untuk melakukan login admin aplikasi web. Tabel 5.5 menyajikan prosedur pengujian melakukan login.

Pengujian dilakukan untuk menguji apakah admin web yang melakukan proses login akan berhasil masuk ke dalam halaman admin dalam aplikasi web.

Tabel 5.5 Pengujian Melakukan Login Admin Aplikasi Web

Test ID	EQ-MW -03
Tujuan Test	Mengecek apakah fungsi melakukan login admin aplikasi web sudah benar.
Kondisi Awal	Pengguna masuk ke halaman login admin di aplikasi web.

Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data Pengguna berupa username adminroot dan password acmilan14.	Pengguna memasukkan username dan password pada kolom yang telah disediakan.	Data yang telah dimasukkan oleh pengguna dicek ke dalam database dan pengguna masuk ke halaman admin di aplikasi web.	Pengguna berhasil masuk kehalaman admin aplikasi web	Proses melakukan login admin aplikasi web berhasil.

LOGIN ADMIN

Username : adminroot

Password :

••••••••

Login

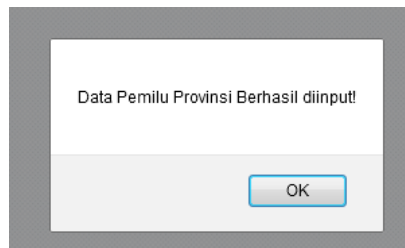
Gambar 5.3 Pengujian Login Admin Web

5.3.1.4. Pengujian Mendaftarkan Pemilu Gubernur

Pada bagian ini dijelaskan tentang pengujian untuk mendaftarkan pemilu gubernur oleh admin di aplikasi web. Tabel 5.6 menyajikan prosedur pengujian mendaftarkan pemilu gubernur. Pengujian dilakukan untuk menguji apakah sistem sudah mampu mencatat detail pemilu gubernur yang dimasukkan oleh admin aplikasi web.

Tabel 5.6 Pengujian Mendaftarkan Pemilu Gubernur

Test ID		EQ-MW -04		
Tujuan Test		Mengecek apakah fungsi mendaftar pemilu gubernur sudah benar.		
Kondisi Awal		Pengguna berada di halaman admin bagian daftar pemilu gubernur .		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data tentang pemilu gubernur berupa provinsi tempat pemilu diselenggarakan di Kalimantan utara, serta nama para pasangan cagub dan cawagub yaitu erlangga & Ibrahim serta nur & wisisono.	<ol style="list-style-type: none"> 1. Admin memilih provinsi di dropdownlist provinsi 2. Admin memasukkan nama pasangan calon. 	Data yang dimasukkan oleh admin diisikan oleh sistem kedalam database.	Data pemilu gubernur tersimpan kedalam database.	Proses mendaftar pemilu gubernur berhasil.

**Gambar 5.4 Pesan Data Pemilu Gubernur Berhasil Didaftarkan**

Selamat Datang : adminroot [Log Out](#)

DAFTAR PEMILU PROVINSI BARU

Nama Pemilu	Pemilihan Umum Kepala Daerah
Pilih Provinsi	Kalimantan Utara
Pasangan Calon ke-1	erlangga & ibrahim
Pasangan Calon ke-2	nur & wisisono
Pasangan Calon ke-3	
Pasangan Calon ke-4	
Pasangan Calon ke-5	

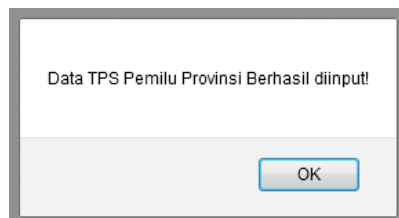
Gambar 5.5 Pengujian Mendaftarkan Pemilu Gubernur

Selamat Datang : adminroot [Log Out](#)

DAFTAR JUMLAH TPS PEMILU PROVINSI

Pilih Pemilu	Pemilihan Umum Kepala Daerah Kalimantan Utara
Pilih Kota/Kabupaten	Kabupaten Bulungan
Jumlah TPS	12

Gambar 5.6 Pengujian Mendaftarkan Jumlah TPS Pemilu Gubernur



Gambar 5.7 Pesan Data Jumlah TPS Pemilu Gubernur Berhasil Dimasukkan

5.3.1.5. Pengujian Mendaftarkan Pemilu Walikota/Bupati

Pada bagian ini dijelaskan tentang pengujian untuk mendaftarkan pemilu walikota/bupati oleh admin di aplikasi web. Tabel 5.7 menyajikan prosedur pengujian mendaftarkan pemilu walikota/bupati. Pengujian dilakukan untuk menguji apakah sistem sudah mampu mencatat detail pemilu walikota/bupati yang dimasukkan oleh admin aplikasi web.

Tabel 5.7 Pengujian Mendaftarkan Pemilu Walikota/Bupati

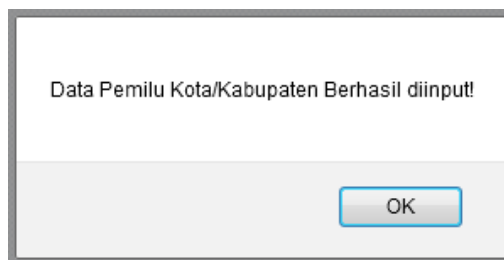
Test ID		EQ-MW -05		
Tujuan Test		Mengecek apakah fungsi mendaftarkan pemilu walikota/bupati sudah benar.		
Kondisi Awal		Pengguna berada di halaman admin bagian daftar pemilu walikota/bupati.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data tentang pemilu walikota/bupati berupa kota/kabupaten dan provinsi tempat pemilu diselenggarakan yaitu di kabupaten Kediri jawa timur, serta nama para pasangan calon yaitu erlangga & Ibrahim serta nur & wisisono.	<ol style="list-style-type: none"> 1. Admin memilih provinsi di dropdownlist provinsi 2. Admin memilih kota di dropdownlist kota 3. admin memasukkan nama pasangan calon. 	Data yang dimasukkan oleh admin diisikan oleh sistem kedalam database.	Data pemilu walikota/bupati tersimpan kedalam database.	Proses mendaftarkan pemilu walikota/bupati berhasil.

Selamat Datang : adminroot [Log Out](#)

DAFTAR PEMILU KOTA/KABUPATEN BARU

Nama Pemilu	<input type="text" value="Pemilihan Umum Kepala Daerah"/>
Pilih Provinsi	<input type="text" value="Jawa Timur"/>
Pilih Kota/Kabupaten	<input type="text" value="Kabupaten Kediri"/>
Pasangan Calon ke-1	<input type="text" value="erlangga & ibrahim"/>
Pasangan Calon ke-2	<input type="text" value="nur & wisisono"/>
Pasangan Calon ke-3	<input type="text"/>
Pasangan Calon ke-4	<input type="text"/>
Pasangan Calon ke-5	<input type="text"/>
Jumlah TPS	<input type="text" value="34"/>
<input type="button" value="Daftar"/>	

Gambar 5.8 Pengujian Mendaftarkan Pemilu Walikota/Bupati



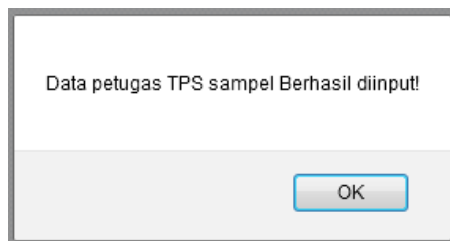
Gambar 5.9 Pesan Data Pemilu Walikota/Bupati Berhasil Didaftarkan

5.3.1.6. Pengujian Mendaftarkan Petugas TPS Sampel

Pada bagian ini dijelaskan tentang pengujian untuk mendaftarkan petugas sampel. Tabel 5.8 menyajikan prosedur pengujian mendaftarkan petugas tps sampel. Pengujian dilakukan untuk menguji apakah sistem sudah mampu untuk mencatat data petugas tps sampel yang dimasukkan oleh admin web.

Tabel 5.8 Pengujian Mendaftarkan Petugas TPS Sampel

Test ID		EQ-MW -06		
Tujuan Test		Mengecek apakah fungsi mendaftarkan petugas tps sampel sudah benar		
Kondisi Awal		Pengguna berada di halaman admin bagian daftar petugas baru		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data berupa username thecomment dan password petugas TPS sampel awawaw	1. Pengguna memilih jenis pemilu 2. pengguna memasukkan username dan password petugas TPS di pemilu tersebut	Data yang dimasukkan oleh admin berupa username dan password petugas TPS sampel tersimpan dalam database	Sistem berhasil menyimpan informasi petugas TPS di dalam database	Proses mendaftarkan petugas TPS sampel berhasil.



Gambar 5.10 Pesan Data Petugas TPS Sampel Berhasil Didaftarkan

Selamat Datang : adminroot [Log Out](#)

DAFTAR PETUGAS BARU

Pilih Pemilu

Pemilihan Umum Kepala Daerah Kalimantan Utara

Pilih Kota/Kabupaten

Kabupaten Bulungan

Username

thecomment

Password

awawaw

Daftar

Gambar 5.11 Pengujian Mendaftarkan Petugas TPS Sampel

5.3.1.7. Pengujian Memilih Tempat Pemilu Gubernur

Pada bagian ini dijelaskan tentang pengujian memilih tempat pemilu gubernur dalam aplikasi web. Tabel 5.9 menyajikan prosedur pengujian memilih tempat pemilu. Pengujian dilakukan untuk menguji apakah sistem sudah mampu mendeteksi tempat pemilu gubernur yang dipilih oleh pengguna.

Tabel 5.9 Pengujian Memilih Tempat Pemilu Gubernur

Test ID		EQ-MW -07		
Tujuan Test		Mengecek apakah fungsi memilih tempat pemilu gubernur sudah benar.		
Kondisi Awal		Pengguna berada di halaman e-quickcount pemilu gubernur.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Nama provinsi yang dipilih oleh pengguna di dropdownlist,	1. Pengguna a memilih salah satu	Nama provinsi yang dipilih oleh pengguna akan dicari id	Sistem berhasil mendeteksi tempat provinsi yang	Proses memilih tempat pemilu gubernur

yaitu provinsi bali	provinsi yang ada pada dropdownlist 2. Pengguna menekan tombol liat hasil	provinsinya oleh sistem di dalam webservice.	dipilih oleh pengguna dan mengambil id provinsi dari database untuk diolah ke proses selanjutnya.	berhasil.
---------------------	--	--	---	-----------

5.3.1.8. Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Gubernur

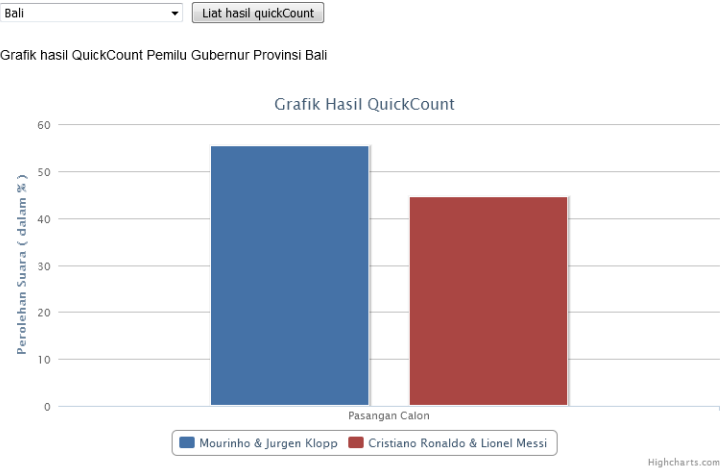
Pada bagian ini dijelaskan tentang pengujian untuk melihat grafik hasil perhitungan cepat pemilu gubernur. Tabel 5.10 menyajikan prosedur pengujian melihat grafik hasil perhitungan cepat pemilu gubernur. Pengujian dilakukan untuk menguji apakah sistem sudah mampu untuk menampilkan grafik yang sesuai dengan pilihan pengguna.

Tabel 5.10 Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Gubernur

Test ID		EQ-MW -08		
Tujuan Test		Mengecek apakah fungsi melihat grafik hasil perhitungan cepat pemilu gubernur sudah benar.		
Kondisi Awal		Pengguna berada pada halaman e-quickcount pemilu gubernur dan menekan tombol liat hasil.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data berupa nama tempat dan id provinsi yang diperoleh dari proses memilih tempat	Pengguna memilih salah satu provinsi yang ada pada dropdownlist lalu menekan	Sistem menampilkan hasil data grafik sesuai dengan provinsi yang dipilih	Sistem berhasil menampilkan grafik yang sesuai dengan pilihan pengguna	Proses menampilkan grafik hasil perhitungan cepat pemilu gubernur berhasil

sebelumnya	tombol liat hasil			
------------	----------------------	--	--	--

Grafik Hasil Quick Count Pemilu Gubernur



Gambar 5.12 Melihat Grafik Hasil Perhitungan Pemilu Gubernur

Tabel hasil QuickCount tiap Kota Pemilu Gubernur Provinsi Bali

Nama Pasangan Calon	Kota	Perolehan Suara (%)
Mourinho & Jurgen Klopp	Kabupaten Badung	42.19
	Kabupaten Bangli	50.20
	Kabupaten Buleleng	72.30
	Kabupaten Gianyar	35.76
	Kabupaten Jembrana	59.89
	Kabupaten Karangasem	68.94
	Kabupaten Klungkung	50.00
	Kabupaten Tabanan	74.31
	Kota Denpasar	48.97

Gambar 5.13 Melihat Tabel Hasil Perhitungan Pemilu Gubernur

5.3.1.9. Pengujian Memilih Tempat Pemilu Walikota/Bupati

Pada bagian ini dijelaskan tentang pengujian memilih tempat pemilu walikota/bupati dalam aplikasi web. Tabel 5.11 menyajikan prosedur pengujian memilih tempat pemilu. Pengujian dilakukan untuk menguji apakah sistem sudah mampu mendeteksi tempat pemilu walikota/bupati yang dipilih oleh pengguna.

Tabel 5.11 Pengujian Memilih Tempat Pemilu Walikota/Bupati

Test ID		EQ-MW -09		
Tujuan Test		Mengecek apakah fungsi memilih tempat pemilu walikota/bupati sudah benar.		
Kondisi Awal		Pengguna berada di halaman e-quickcount pemilu walikota/gubernur.		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Nama kota/kabupaten serta provinsi yang dipilih oleh pengguna di dropdownlist yaitu kota Surabaya Jawa timur.	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu provinsi yang ada pada dropdownlist 2. Pengguna memilih kota/kabupaten yang ada pada provinsi tersebut pada dropdownlist 	Nama provinsi serta kota yang dipilih oleh pengguna akan dicari id provinsinya dan id kotanya oleh sistem di dalam webservice.	Sistem berhasil mendeteksi tempat kota/kabupaten yang dipilih oleh pengguna dan mengambil id kota dari database untuk diolah ke proses selanjutnya.	Proses memilih tempat pemilu walikota/bupati berhasil.

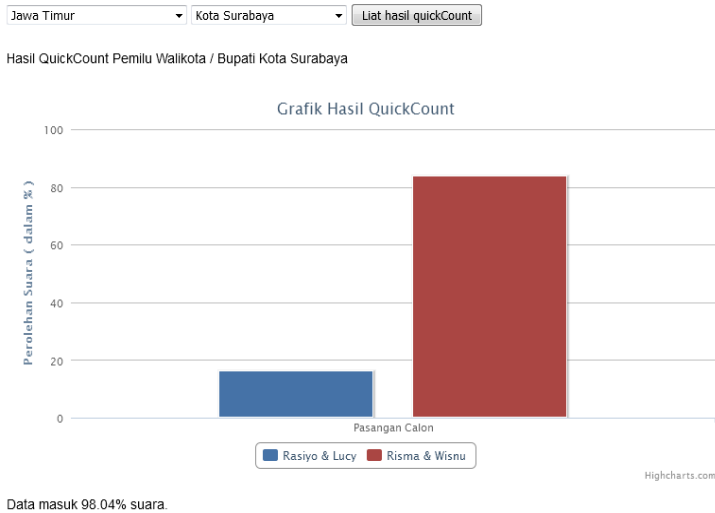
5.3.1.10. Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Walikota/Bupati

Pada bagian ini dijelaskan tentang pengujian untuk melihat grafik hasil perhitungan cepat pemilu walikota/bupati. Tabel 5.12 menyajikan prosedur pengujian melihat grafik hasil perhitungan cepat pemilu walikota/bupati. Pengujian dilakukan untuk menguji apakah sistem sudah mampu untuk menampilkan grafik yang sesuai dengan pilihan pengguna.

Tabel 5.12 Pengujian Melihat Grafik Hasil Perhitungan Cepat Pemilu Walikota/Bupati

Test ID		EQ-MW -10		
Tujuan Test		Mengecek apakah fungsi melihat info poin pengguna sudah benar.		
Kondisi Awal		Pengguna melakukan klik tombol panel kanan yang terdapat pada konten <i>header</i>		
Data Input	Prosedur pengujian	Hasil yang diharapkan	Hasil yang diperoleh	Kesimpulan
Data berupa nama tempat dan id kota yang diperoleh dari proses memilih tempat sebelumnya yaitu kota surabaya	<ol style="list-style-type: none"> 1. Pengguna memilih salah satu provinsi 2. Pengguna memilih kota/kabupaten yang ada di dalam provinsi tersebut pada dropdown list 3. Pengguna menekan tombol liat hasil 	Sistem menampilkan hasil data grafik sesuai dengan kota yang dipilih	Sistem berhasil menampilkan grafik yang sesuai dengan pilihan pengguna	Proses menampilkan grafik hasil perhitungan cepat pemilu walikota/bupati berhasil

Grafik Hasil Quick Count Pemilu Walikota/Bupati



Gambar 5.14 Pengujian Melihat Grafik Hasil Perhitungan Pemilu Walikota/Bupati

5.4 Evaluasi Pengujian

Dari skenario pengujian kebutuhan fungsional dan kebutuhan non-fungsional yang telah dilakukan pada subbab 5.2, dapat diberikan paparan evaluasi sebagai berikut.

1. Proses Melakukan Login Aplikasi Mobile telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-01], dimana detail pengujian dipaparkan secara rinci.
2. Proses Mengunggah Nilai Suara Calon telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-02], dimana detail pengujian dipaparkan secara rinci.

3. Proses Melakukan Login Admin Aplikasi Web telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-03], dimana detil pengujian dipaparkan secara rinci.
4. Proses Mendaftarkan Pemilu Gubernur telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-04], dimana detil pengujian dipaparkan secara rinci.
5. Proses Mendaftarkan Pemilu Walikota/Bupati telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-05], dimana detil pengujian dipaparkan secara rinci.
6. Proses Mendaftarkan Petugas TPS Sampel telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-06], dimana detil pengujian dipaparkan secara rinci.
7. Proses Memilih Tempat Pemilu Gubernur telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-07], dimana detil pengujian dipaparkan secara rinci.
8. Proses Melihat Grafik Hasil Perhitungan Cepat Pemilu Gubernur telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-08], dimana detil pengujian dipaparkan secara rinci.
9. Proses Memilih Tempat Pemilu Walikota/Bupati telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-09], dimana detil pengujian dipaparkan secara rinci.

10. Proses Melihat Grafik Hasil Perhitungan Cepat Pemilu Walikota/Bupati telah berhasil dan berjalan seperti yang diharapkan. Hal ini terlihat pada pengujian [EQ-MW-10], dimana detil pengujian dipaparkan secara rinci.
11. Semua proses berjalan dengan menggunakan jaringan internet, sehingga performa semua proses tergantung dari cepatnya koneksi internet yang digunakan oleh pengguna.

5.5 Pengujian Langsung

Pengujian langsung dari aplikasi mobile e-QuickCount ini telah dilakukan pada Pemilu Serentak 9 Desember 2015 tepatnya pada Pemilu Walikota Surabaya.

Untuk TPS sampel berjumlah 4, diambil dari TPS disekitar kampus ITS, yaitu di Keputih gang pasar (Gambar 5.15), perumdos blok J (Gambar 5.16), perumdos blok T (Gambar 5.17) dan perumdos blok U (Gambar 5.18).

Skenario pengujian yaitu menempatkan 1 orang petugas di masing-masing TPS sampel. Setiap petugas telah dibekali prosedur penggunaan aplikasi mobile e-QuickCount, ID, username, serta password untuk masuk kedalam aplikasi.

Para petugas telah standby di TPS sejak perhitungan suara dimulai, yaitu pukul 13.00 WIB hingga perhitungan selesai. Setelah proses perhitungan selesai, petugas membuka aplikasi e-QuickCount yang telah terinstal di ponsel pintar masing-masing, lalu login memasukkan ID, username serta password, setelah itu memasukkan nilai atau perolehan suara para pasangan calon dan mengunggahnya ke server e-QuickCount melalui jaringan internet (Gambar 5.19).

Proses pengujian langsung ini berjalan dengan lancar yaitu setiap petugas telah berhasil login dan nilai atau suara para pasangan calon telah berhasil diunggah ke webservice dan tersimpan di dalam database yang ada pada server.



Gambar 5.15 Kondisi TPS Keputih Gang Pasar



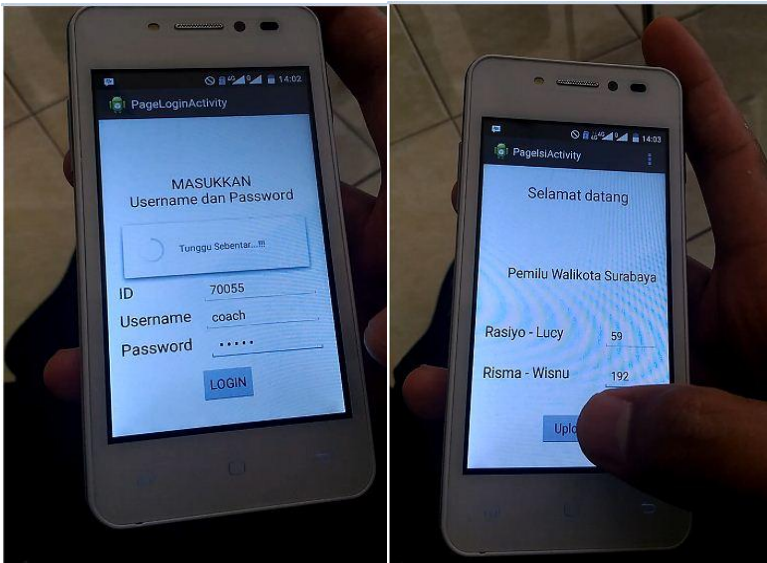
Gambar 5.16 Kondisi TPS Perumdos Blok J



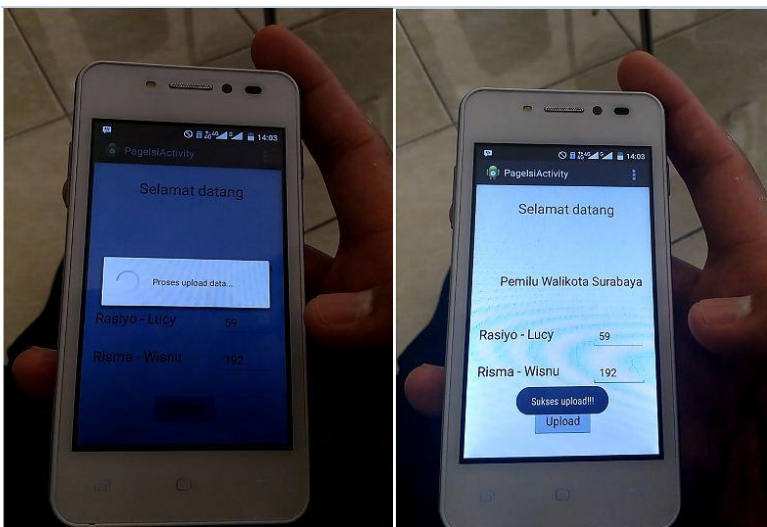
Gambar 5.17 Kondisi TPS Perumdos Blok T



Gambar 5.18 Kondisi TPS Perumdos Blok U



Gambar 5.19 Implementasi Login dan Isi Data



Gambar 5.20 Implementasi Unggah Data

BAB VI PENUTUP

Bab ini membahas kesimpulan akhir dan saran yang diperoleh selama proses perancangan e-QuickCount.

6.1. Kesimpulan

Dalam proses pengerjaan Tugas Akhir dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut.

1. Sistem sudah mampu dijalankan pada alat komunikasi bergerak dengan sistem operasi android.
2. Sistem sudah mampu menampilkan segala macam informasi tentang perhitungan cepat atau quick count beserta hasilnya pada aplikasi web.(UC 5.3.1.8)
3. Sistem sudah mampu mengambil dan menampilkan database yang di simpan dalam server pada aplikasi mobile maupun web.(UC 5.3.1.2)
4. Sistem sudah mampu membuat webservice untuk menangani proses perhitungan suara, mengolahnya hingga menampilkannya dalam bentuk grafik pada aplikasi web.(UC 5.3.1.8)
5. Sistem sudah mampu menerapkan pertukaran data berupa JSON pada aplikasi mobile.
6. Sistem sudah mampu memproses perhitungan cepat pemilukada dengan maksimal 5 pasangan calon di satu pemilu.
7. Sistem sudah mampu mengupdate database yang ada pada server melalui aplikasi mobile maupun web.(UC 5.3.1.2)
8. Sistem sudah mampu memberikan pilihan kota/kabupaten yang sesuai dengan provinsi pilihan pengguna pada dropdownlist di aplikasi web.(UC 5.3.1.9)

9. Sistem sudah mampu menampilkan grafik sesuai dengan pilihan pengguna pada aplikasi web.(UC 5.3.1.9)
10. Sistem sudah mampu mengintegrasikan PHP, Mysql serta Java dalam proses pembuatan aplikasi mobile serta web dengan baik.
11. Sistem mampu menangani semua proses quick count dari mulai menghitung perolehan suara hingga menampilkannya ke media berupa internet, sehingga waktu pelaksanaan proses perhitungan cepat atau quick count dapat dipersingkat.

6.2. Saran

Saran penulis terhadap proses pengembangan aplikasi e-QuickCount adalah sebagai berikut.

1. Aplikasi ini kedepannya dapat dikembangkan tidak terbatas pada perhitungan cepat atau quick count namun juga perhitungan total atau real count.
2. Aplikasi ini dibangun tanpa menggunakan suatu framework, untuk kedepannya dapat menggunakan framework tertentu agar performa maupun tampilannya lebih dinamis.
3. Untuk keamanan sistem, pada tiap-tiap halaman perlu dilakukan pengecekan pengguna sehingga meningkatkan keamanan sistem.

LAMPIRAN A IMPLEMENTASI

Kode Sumber A. 1 Implementasi Pertukaran Data

JSONParser.java

```
public JSONObject getJSONFromUrl(String url) {  
  
    try {  
        DefaultHttpClient httpClient = new  
DefaultHttpClient();  
        HttpPost httpPost = new HttpPost(url);  
        HttpResponse httpResponse =  
httpClient.execute(httpPost);  
        HttpEntity httpEntity = httpResponse.getEntity();  
        is = httpEntity.getContent();  
    } catch (UnsupportedEncodingException e) {  
        e.printStackTrace();  
    } catch (ClientProtocolException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    try {  
        BufferedReader reader = new BufferedReader(new  
InputStreamReader(  
            is, "iso-8859-1"), 8);  
        StringBuilder sb = new StringBuilder();  
        String line = null;  
        while ((line = reader.readLine()) != null) {  
            sb.append(line + "\n");  
        }  
        is.close();  
        json = sb.toString();  
    } catch (Exception e) {  
        Log.e("Buffer Error", "Error converting result  
" + e.toString());  
    }  
    try {  
        jsonObj = new JSONObject(json);  
    } catch (JSONException e) {  
        Log.e("JSON Parser", "Error parsing data " +  
e.toString());  
    }  
    return jsonObj;  
}
```

```

public JSONObject makeHttpRequest(String url, String method,
    List<NameValuePair> params) {

    try {
        if (method == "POST") {
            DefaultHttpClient httpClient = new
            DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);
            httpPost.setEntity(new
            UriEncodedFormEntity(params));

            HttpResponse httpResponse =
            httpClient.execute(httpPost);
            HttpEntity httpEntity =
            httpResponse.getEntity();
            is = httpEntity.getContent();

        } else if (method == "GET") {
            // request method is GET
            DefaultHttpClient httpClient = new
            DefaultHttpClient();
            String paramString =
            URLEncodedUtils.format(params, "utf-8");
            url += "?" + paramString;
            HttpGet httpGet = new HttpGet(url);

            HttpResponse httpResponse =
            httpClient.execute(httpGet);
            HttpEntity httpEntity =
            httpResponse.getEntity();
            is = httpEntity.getContent();
        }

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```
try {
    BufferedReader reader = new BufferedReader(new
InputStreamReader(
        is, "iso-8859-1"), 8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    is.close();
    json = sb.toString();
} catch (Exception e) {
    Log.e("Buffer Error", "Error converting result " +
e.toString());
}

// try parse the string to a JSON object
try {
    jsonObj = new JSONObject(json);
} catch (JSONException e) {
    Log.e("JSON Parser", "Error parsing data " +
e.toString());
}

// return JSON String
return jsonObj;
}
```

SessionManager.java

```

public SessionManager(Context context){
    this._context = context;
    pref = _context.getSharedPreferences(PREF_NAME,
    PRIVATE_MODE);
    editor = pref.edit();
}

public void createLoginSession(String id_petugas, String
nama, String nama_pemilu,
    String calon1, String calon2, String calon3,
String calon4, String calon5,
    String idcalon1, String idcalon2, String
idcalon3, String idcalon4,
    String idcalon5, String idkota){
    editor.putString(KEY_IDPETUGAS, id_petugas);
    editor.putString(KEY_USERNAME, nama);
    editor.putString(KEY_NAMAPEMILU, nama_pemilu);
    editor.putString(KEY_CALON1, calon1);
    editor.putString(KEY_CALON2, calon2);
    editor.putString(KEY_CALON3, calon3);
    editor.putString(KEY_CALON4, calon4);
    editor.putString(KEY_CALON5, calon5);
    editor.putString(KEY_IDCALON1, idcalon1);
    editor.putString(KEY_IDCALON2, idcalon2);
    editor.putString(KEY_IDCALON3, idcalon3);
    editor.putString(KEY_IDCALON4, idcalon4);
    editor.putString(KEY_IDCALON5, idcalon5);
    editor.putString(KEY_IDKOTA, idkota);
    editor.commit();
}

public void logoutUser(){
    // Clearing all data from Shared Preferences
    editor.clear();
    editor.commit();

    Intent i = new Intent(_context,
PageLoginActivity.class);
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    _context.startActivity(i);
}

```



```
public HashMap<String, String> getUserDetails(){
    HashMap<String, String> user = new HashMap<String,
String>();

    user.put(KEY_IDPETUGAS, pref.getString(KEY_IDPETUGAS,
null));
    user.put(KEY_USERNAME, pref.getString(KEY_USERNAME,
null));
    user.put(KEY_NAMAPEMILU,
pref.getString(KEY_NAMAPEMILU, null));
    user.put(KEY_CALON1, pref.getString(KEY_CALON1,
null));
    user.put(KEY_CALON2, pref.getString(KEY_CALON2,
null));
    user.put(KEY_CALON3, pref.getString(KEY_CALON3,
null));
    user.put(KEY_CALON4, pref.getString(KEY_CALON4,
null));
    user.put(KEY_CALON5, pref.getString(KEY_CALON5,
null));
    user.put(KEY_IDCALON1, pref.getString(KEY_IDCALON1,
null));
    user.put(KEY_IDCALON2, pref.getString(KEY_IDCALON2,
null));
    user.put(KEY_IDCALON3, pref.getString(KEY_IDCALON3,
null));
    user.put(KEY_IDCALON4, pref.getString(KEY_IDCALON4,
null));
    user.put(KEY_IDCALON5, pref.getString(KEY_IDCALON5,
null));
    user.put(KEY_IDKOTA, pref.getString(KEY_IDKOTA,
null));

    return user;
}
```

PageIsiActivity.java

```

protected String doInBackground(String... arg0) {
    String strNIL1 =
    nilaicalon1.getText().toString();
    String strNIL2 =
    nilaicalon2.getText().toString();
    String strNIL3 =
    nilaicalon3.getText().toString();
    String strNIL4 =
    nilaicalon4.getText().toString();
    String strNIL5 =
    nilaicalon5.getText().toString();
    HashMap<String, String> user2 =
    session.getUserDetails();
    String idp =
    user2.get(SessionManager.KEY_IDPETUGAS);
    String idcalon1 =
    user2.get(SessionManager.KEY_IDCALON1);
    String idcalon2 =
    user2.get(SessionManager.KEY_IDCALON2);
    String idcalon3 =
    user2.get(SessionManager.KEY_IDCALON3);
    String idcalon4 =
    user2.get(SessionManager.KEY_IDCALON4);
    String idcalon5 =
    user2.get(SessionManager.KEY_IDCALON5);
    String idkota =
    user2.get(SessionManager.KEY_IDKOTA);

    List<NameValuePair> params = new
    ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("nilai1",
    strNIL1));
    params.add(new BasicNameValuePair("nilai2",
    strNIL2));
    params.add(new BasicNameValuePair("nilai3",
    strNIL3));
    params.add(new BasicNameValuePair("nilai4",
    strNIL4));
    params.add(new BasicNameValuePair("nilai5",
    strNIL5));
    params.add(new BasicNameValuePair("idcalon1",
    idcalon1));

```

```

params.add(new BasicNameValuePair("idcalon2", idcalon2));
params.add(new BasicNameValuePair("idcalon3",
idcalon3));
params.add(new BasicNameValuePair("idcalon4",
idcalon4));
params.add(new BasicNameValuePair("idcalon5",
idcalon5));
params.add(new BasicNameValuePair("idp", idp));
params.add(new BasicNameValuePair("idkota",
idkota));

JSONObject json = jParser.makeHttpRequest(url,
"POST", params);

try {
    success = json.getString("success");
} catch (Exception e) {
    Toast.makeText(getApplicationContext(),
"Error",
        Toast.LENGTH_LONG).show();
}
return null;
}

protected void onPostExecute(String file_url) {
    // dismiss the dialog once done
    pDialog.dismiss();

    if (success.equals("1"))
    {
        Toast.makeText(getApplicationContext(),
"Sukses upload!!!", Toast.LENGTH_LONG).show();
        submit = (Button)findViewById(R.id.upload);
        submit.setClickable(false);
        nilaicalon1 = (EditText)
findViewById(R.id.nilaicalon1);
        nilaicalon2 = (EditText)
findViewById(R.id.nilaicalon2);
        nilaicalon3 = (EditText)
findViewById(R.id.nilaicalon3);
        nilaicalon4 = (EditText)
findViewById(R.id.nilaicalon4);
        nilaicalon4 = (EditText)
findViewById(R.id.nilaicalon5);
    }
}

```

```
nilaicalon1.setEnabled(false);  
    nilaicalon2.setEnabled(false);  
    nilaicalon3.setEnabled(false);  
    nilaicalon4.setEnabled(false);  
    nilaicalon5.setEnabled(false);  
}  
else  
{  
    Toast.makeText(getApplicationContext(),  
    "Gagal upload!!!", Toast.LENGTH_LONG).show();  
}  
}
```


Kode Sumber A. 2 Implementasi Login Aplikasi Mobile

Loginhp.php

```

$id_petugas = $_GET["id_petugas"];
$username   = $_GET["username"];
$password   = $_GET["password"];
$query      = "select * from petugas where
id_petugas='$id_petugas' and username='$username' and
password='$password'";
$queryflag  = "select * from petugas where
id_petugas='$id_petugas' and username='$username' and
password='$password'";
$hasil      = mysql_query($query);
$hasil2     = mysql_query($queryflag);
$flagpetugas = mysql_fetch_array($hasil2);
$flag       = $flagpetugas['flag_petugas'];
$idlist     = $flagpetugas['id_list'];
if (mysql_num_rows($hasil) > 0 && $flag == 0) {
    $response = array();
    $response["loginhp"] = array();
    $query_judul = "select nama_pemilu from `list pemilu`
where id_list='$idlist' ";
    $hasiljudul = mysql_query($query_judul);
    $judul      = mysql_fetch_array($hasiljudul);
    $h['nama_pemilu'] = $judul['nama_pemilu'];
    $querycalon  = "select id_calon from `info peserta`
where id_list='$idlist' ";
    $hasilcalon = mysql_query($querycalon);
    $calon      = mysql_fetch_array($hasilcalon);
    $jumlahcalon = mysql_num_rows($hasilcalon);
    if ($jumlahcalon == 1)
    {
        $querycalonall = "select id_calon from `info
peserta` where id_list='$idlist' ";
        $hasilcalonall = mysql_query($querycalonall);
        $calonall      =
mysql_fetch_array($hasilcalonall);
        $idcalon = $calonall['id_calon'];
        $h['idcalon1'] = $idcalon;
        $h['idcalon2'] = 0;
        $h['idcalon3'] = 0;
        $h['idcalon4'] = 0;
        $h['idcalon5'] = 0;
    }
}

```

```

$querycalon1 = "select nama_calon from `info peserta` where
id_list='$idlist' ";

$hasilcalon1 = mysql_query($querycalon1);
$calon1 = mysql_fetch_array($hasilcalon1);
$h['calon1'] = $calon1['nama_calon'];
$h['calon2'] = " ";
$h['calon3'] = " ";
$h['calon4'] = " ";
$h['calon5'] = " ";
}
.
.
.
.
.
.
.
.
while ($data = mysql_fetch_array($hasil))
{
    $h['id_petugas'] = $data['id_petugas'];
    $h['id_list'] = $data['id_list'];
    $h['id_kota'] = $data['id_kota'];
    $h['username'] = $data['username'];
    $h['password'] = $data['password'];
    $h['flag_petugas'] = $data['flag_petugas'];

    array_push($response["loginhp"], $h);
}
$response["success"] = "1";
echo json_encode($response);
}
if (mysql_num_rows($hasil) > 0 && $flag == 1) {
    $response["success"] = "2";
    //$response["message"] = "Tidak ada data";
    echo json_encode($response);
}
else {
    $response["success"] = "0";
    $response["message"] = "Tidak ada data";
    echo json_encode($response);
}
}

```

Kode Sumber A. 3 Implementasi Menampilkan Keterangan Pemilu

PageIsiActivity.java

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_page_isi);

    session = new SessionManager(getApplicationContext());
    /*Toast.makeText(getApplicationContext(),
        "User Login Status: " + session.isLoggedIn(),
        Toast.LENGTH_LONG)
        .show();*/

    //session.checkLogin();
    HashMap<String, String> user2 =
    session.getUserDetails();

    nama = user2.get(SessionManager.KEY_USERNAME);
    TextView status = (TextView)
    findViewById(R.id.status);
    status.setText(Html.fromHtml("Selamat datang <b>" +
    nama + "</b> "));

    nama_pem = user2.get(SessionManager.KEY_NAMAPEMILU);
    TextView judul = (TextView) findViewById(R.id.judul);
    judul.setText(Html.fromHtml(nama_pem));

    calon1 = user2.get(SessionManager.KEY_CALON1);
    TextView namacalon1 = (TextView)
    findViewById(R.id.namacalon1);
    namacalon1.setText(Html.fromHtml(calon1));

    calon2 = user2.get(SessionManager.KEY_CALON2);
    TextView namacalon2 = (TextView)
    findViewById(R.id.namacalon2);
    namacalon2.setText(Html.fromHtml(calon2));

    nilaicalon2 = (EditText)
    findViewById(R.id.nilaicalon2);
    nilaicalon2.setVisibility(View.INVISIBLE);
    int panjangcalon2 = calon2.length();
    if (panjangcalon2 > 1) {
        nilaicalon2.setVisibility(View.VISIBLE);
    }
}
```

```

calon3 = user2.get(SessionManager.KEY_CALON3);
TextView namacalon3 = (TextView)
findViewById(R.id.namacalon3);
namacalon3.setText(Html.fromHtml(calon3));

nilaicalon3 = (EditText)
findViewById(R.id.nilaicalon3);
nilaicalon3.setVisibility(View.INVISIBLE);
int panjangcalon3 = calon3.length();
if (panjangcalon3 > 1) {
nilaicalon3.setVisibility(View.VISIBLE);
}

calon4 = user2.get(SessionManager.KEY_CALON4);
TextView namacalon4 = (TextView)
findViewById(R.id.namacalon4);
namacalon4.setText(Html.fromHtml(calon4));

nilaicalon4 = (EditText)
findViewById(R.id.nilaicalon4);
nilaicalon4.setVisibility(View.INVISIBLE);
int panjangcalon4 = calon4.length();
if (panjangcalon4 > 1) {
nilaicalon4.setVisibility(View.VISIBLE);
}

calon5 = user2.get(SessionManager.KEY_CALON5);
TextView namacalon5 = (TextView)
findViewById(R.id.namacalon5);
namacalon5.setText(Html.fromHtml(calon5));

nilaicalon5 = (EditText)
findViewById(R.id.nilaicalon5);
nilaicalon5.setVisibility(View.INVISIBLE);
int panjangcalon5 = calon5.length();
if (panjangcalon5 > 1) {
nilaicalon5.setVisibility(View.VISIBLE);
}

nilaicalon1
(EditText)findViewById(R.id.nilaicalon1);

```


Kode Sumber A. 4 Implementasi Unggah Data Nilai Suara Calon

Insertnilai.php

```

$Nilai1    = $_POST['nilai1'];
$Nilai2    = $_POST['nilai2'];
$Nilai3    = $_POST['nilai3'];
$Nilai4    = $_POST['nilai4'];
$Nilai5    = $_POST['nilai5'];
$idcalon1  = $_POST['idcalon1'];
$idcalon2  = $_POST['idcalon2'];
$idcalon3  = $_POST['idcalon3'];
$idcalon4  = $_POST['idcalon4'];
$idcalon5  = $_POST['idcalon5'];
$IDP       = $_POST['idp'];
$idkota    = $_POST['idkota'];

header('Content-Type: text/xml');

$query_ambil_calon1 = "select nilai from calon where
id_calon='$idcalon1' and id_kota='$idkota' ";
$hasil1 = mysql_query($query_ambil_calon1);
$data1 = mysql_fetch_array($hasil1);
$Nilai1 = (int)$Nilai1 + (int)$data1['nilai'];

$query_ambil_calon2 = "select nilai from calon where
id_calon='$idcalon2' and id_kota='$idkota' ";
$hasil2 = mysql_query($query_ambil_calon2);
$data2 = mysql_fetch_array($hasil2);
$Nilai2 = (int)$Nilai2 + (int)$data2['nilai'];

.
.
.
.
.
$query = "update calon set nilai = '$Nilai1' where
id_calon='$idcalon1' and id_kota='$idkota' ";
$hasil = mysql_query($query);
$query2 = "update calon set nilai = '$Nilai2' where
id_calon='$idcalon2' and id_kota='$idkota' ";
$hasil2 = mysql_query($query2);

.
.
.
.

```

```
if($hasil)
{
    $response["success"] = "1";
    $response["message"] = "Data sukses diinput";
    echo json_encode($response);
}
else
{
    $response["success"] = "0";
    $response["message"] = "Maaf , terjadi kesalahan";

    // echoing JSON response
    echo json_encode($response);
}
```

Kode Sumber A. 5 Implementasi Memilih Tempat Pemilu

Quickcount.php

```

$quer2="SELECT nama_propinsi,id_propinsi FROM propinsi order
by nama_propinsi";
if(isset($cat) and
strlen($cat) > 0){
    $quer="SELECT
nama_kota FROM kota where id_propinsi=$cat order by
nama_kota";
}
else{
    $quer="SELECT
nama_kota FROM kota order by nama_kota";
}
echo "<form method=post
name=f1 action=''>";
echo "<select
name='cat' onchange=\"reload(this.form)\"><option
value=''>Pilih Provinsi</option>";
foreach
($dbo->query($quer2) as $noticia2) {
    if($noticia2['id_propinsi']==@$cat){
        echo "<option selected
value='$noticia2[id_propinsi]'>$noticia2[nama_propinsi]</opti
on>". "<BR>";
        $prop=$_POST['cat'];
    }
    else
    {
        echo "<option
value='$noticia2[id_propinsi]'>$noticia2[nama_propinsi]</opti
on>";
        $prop=$_POST['cat'];
    }
}
echo "</select>";
echo "&nbsp;";
echo "<select
name='subcat'><option value=''>Pilih
Kota/Kabupaten</option>";

```

```

foreach ($dbo->query($quer) as $noticia) {
    echo
    "<option
    value='$noticia[nama_kota]'">$noticia[nama_kota]</option>";
    $kot=$_POST['subcat'];
    }
    echo "</select>";
    echo "&nbsp;";
    echo "<input
    type=submit name='liathasil' value='Liat hasil quickCount'>";
    echo "</form>";
    .
    .
    .
    .
    .
    if ((isset($_POST['liathasil'])) AND ($_POST['cat'] <> ""))
    AND ($_POST['subcat'] <> "")) {
        $searchprop =
        $_POST['cat'];
        $searchkota =
        $_POST['subcat'];
        $sqlcekidkota =
        mysql_query("SELECT * FROM kota where nama_kota='$searchkota'
        ") or die (mysql_error());
        $dataidkota =
        mysql_fetch_array($sqlcekidkota);
        $idkota =
        $dataidkota['id_kota'];
        $sqlcekpemilukota =
        mysql_query("SELECT * FROM `pemilu kota` where
        id_kota='$idkota'") or die (mysql_error());
        $dataflagkota =
        mysql_fetch_array($sqlcekpemilukota);
        $flagkota =
        $dataflagkota['flag_kota'];
        $adapemilukota =
        mysql_num_rows($sqlcekpemilukota);
        .
        .
        .
    }
}

```


Kode Sumber A. 6 Implementasi Menampilkan Grafik Pemilu Kabupaten

Quickcount.php

```

if ((isset($_POST['liathasil'])) AND ($_POST['cat'] <> "")
AND ($_POST['subcat'] <> ""))
{
    $searchkota
= $_POST['subcat'];

    $sqlcekidkota = mysql_query("SELECT * FROM kota
where nama_kota='$searchkota' ") or die (mysql_error());
    $dataidkota
= mysql_fetch_array($sqlcekidkota);
    $idkota =
$dataidkota['id_kota'];

    $sqlcekpemilukota = mysql_query("SELECT * FROM
`pemilu_kota` where id_kota='$idkota'") or die
(mysql_error());

    $datapemilukota
=
mysql_fetch_array($sqlcekpemilukota);
    $flagkota =
$datapemilukota['flag_kota'];

    $adapemilukota = mysql_num_rows($sqlcekpemilukota);
    if
($adapemilukota > 0 && $flagkota == 1) {
        $idpemilukota = $datapemilukota['id_list'];

        $sql = mysql_query("SELECT * FROM `info peserta`
where id_list='$idpemilukota'") or die (mysql_error());

        $sqltotal = mysql_query("SELECT SUM(nilai) AS
total_suara from calon where id_list='$idpemilukota'")
or
die (mysql_error());

        $totalnilai = mysql_fetch_array($sqltotal);
        $nilai = (float)$totalnilai['total_suara'];
    }
}

```

```

while ($data = mysql_fetch_array($sql)) {

    $namacalon = $data['nama_calon'];
    $idcalon = $data['id_calon'];

    $sqljumlahsuara = mysql_query("SELECT nilai FROM calon
    WHERE id_calon='$idcalon'") or die (mysql_error());

    while ($datajumlah = mysql_fetch_array($sqljumlahsuara)) {

        $jumlah = $datajumlah['nilai'];

    }

    {
        name: '<?php echo $namacalon; ?>',
        data: [<?php
            echo
            number_format($jumlah/$nilai*100,2);
        ?>]
    },
    },
    <?php
    }
    } else {
        //
        echo 'Maaf,
        menampilkan pesan zero data
        tidak ada pemilu saat ini.'; }
    }
}

```

Kode Sumber A. 7 Implementasi Menampilkan Grafik Pemilu Provinsi

Quickcountprov.php

```

if ((isset($_POST['liathasil'])) AND ($_POST['cat'] <> ""))//
AND ($_POST['subcat'] <> "")
{
    $searchprovinsi = $_POST['cat'];

    $sqlcekpemiluprov2 = mysql_query("SELECT * FROM
`pemilu propinsi` where id_propinsi='$searchprovinsi'")
or die
(mysql_error());

    $datapemiluprov2 =
mysql_fetch_array($sqlcekpemiluprov2);
    $flagprov2 =
    $datapemiluprov2['flag_propinsi'];

    $adapemiluprov2 = mysql_num_rows($sqlcekpemiluprov2);
    if
    ($adapemiluprov2 > 0 && $flagprov2 == 1) {

        $idpemiluprov2 = $datapemiluprov2['id_list'];

        $sqlnama2 = mysql_query("SELECT * FROM `info peserta`
where id_list='$idpemiluprov2'") or die (mysql_error());

        $sqltotal2 = mysql_query("SELECT SUM(nilai) AS
total_suara2 from calon where id_list='$idpemiluprov2'")
or
die (mysql_error());

        $totalnilai2 = mysql_fetch_array($sqltotal2);

        $nilai2 = (float)$totalnilai2['total_suara2'];
        while
        ($data2 = mysql_fetch_array($sqlnama2)) {
            $namacalon2 = $data2['nama_calon'];
            $idcalon2 =
            $data2['id_calon'];
            $sqljumlahsuara2 =
            mysql_query("SELECT SUM(nilai) AS total_suara FROM calon WHERE
id_calon='$idcalon2'
and id_list='$idpemiluprov2'") or die
(mysql_error());

```

```

while ($datajumlah2 = mysql_fetch_array($sqljumlahsuara2)) {
    $jumlah2 = $datajumlah2['total_suara'];
}
?>
{
    name: '<?php echo $namacalon2; ?>',
    data: [<?php
        echo
        number_format($jumlah2/$nilai2*100,2);
    ?>]
},
<?php
}
} else {
    //
    echo 'Maaf,
menampilkan pesan zero data
tidak ada pemilu saat ini.'; }
}

```


Kode Sumber A. 8 Implementasi Menghitung Prosentase Nilai

Quickcount.php

```

if ($adapemilukota > 0 && $flagkota == 1) {
    QuickCount      Pemilu      Walikota      echo      "Hasil
    ".$dataaidkota['nama_kota']. '<br>';      /      Bupati
    echo      "<div
    id='container' ></div>". '<br>';
    $idpemukota
    = $dataflagkota['id_list'];
    $sqlcektotpet
    = mysql_query("SELECT * FROM petugas where
    id_list='".$idpemukota'") or die (mysql_error());
    $totalpetugas
    = mysql_num_rows($sqlcektotpet);
    $sqlcektotpetupload = mysql_query("SELECT * FROM
    petugas where id_list='".$idpemukota' and flag_petugas=1 ")
    or die
    (mysql_error());
    $totalpetugasupload
    =
    mysql_num_rows($sqlcektotpetupload);
    $prosentase =
    number_format($totalpetugasupload/$totalpetugas*100,2);
    echo      "Data
    masuk ".$prosentase."% suara.";
    }

```

Kode Sumber A. 9 Implementasi Login Admin Aplikasi Web

login.php

```

session_start();
$error='';
if (isset($_POST['submit'])) {
if (empty($_POST['username']) || empty($_POST['password'])) {
$error = "Username or Password is invalid";
}
else
{
$username=$_POST['username'];
$password=$_POST['password'];
$connection = mysql_connect("localhost", "root", "");
$username = stripslashes($username);
$password = stripslashes($password);
$username = mysql_real_escape_string($username);
$password = mysql_real_escape_string($password);
$db = mysql_select_db("equickcount", $connection);
$query = mysql_query("select * from admin where
password='$password' AND username='$username'", $connection);
$rows = mysql_num_rows($query);
if ($rows == 1) {
$_SESSION['login_user']=$username;
header("location: admin.php");
} else {
$error = "Username atau Password belum terdaftar";
}
mysql_close($connection);
}
}

```

Session.php

```

session_start();

$user_check=$_SESSION['login_user'];

$ses_sql=mysql_query("select username from admin where
username='$user_check'", $connection);
$row = mysql_fetch_assoc($ses_sql);
$login_session =$row['username'];
if(!isset($login_session)){
mysql_close($connection);
header('Location: adminlogin.php');
}

```

Kode Sumber A. 10 Implementasi Mendaftar Pemilu Baru

Action_inputpemilukota.php

```

$namakota=$_POST['kotapemilu'];
$sqlgetidkota = mysql_query("SELECT * FROM kota where
nama_kota='$namakota' ") or die (mysql_error());
$dataidkota = mysql_fetch_array($sqlgetidkota);
$idkota = $dataidkota['id_kota'];

$namaprovinsi=$_POST['provinsipemilu'];
$sqlgetidprov = mysql_query("SELECT * FROM propinsi
where nama_propinsi='$namaprovinsi' ") or die (mysql_error());
$dataidprov = mysql_fetch_array($sqlgetidprov);
$idprov = $dataidprov['id_propinsi'];

//$namapemilu = $_POST['nama_pemilu'];
$namapemilu = "Pemilihan Umum Kepala Daerah
".$namakota;

$sqlinsert_listpemilu = mysql_query("insert into
`list_pemilu` (nama_pemilu, flag_pemilu) values
('$namapemilu', 0)");
$insert1=1;

$sqlgetidlist = mysql_query("SELECT id_list FROM
`list_pemilu` where nama_pemilu='$namapemilu' and
flag_pemilu=0 ") or die (mysql_error());
$dataidlist = mysql_fetch_array($sqlgetidlist);
$idlist = $dataidlist['id_list'];

$sqlinsert_pemilukota = mysql_query("insert into
`pemilu kota` (id_kota, id_list, flag_kota) values ('$idkota',
'$idlist', 0)");
$insert2=1;

$namacalon1 = $_POST['calon1'];
$namacalon2 = $_POST['calon2'];
$namacalon3 = $_POST['calon3'];
$namacalon4 = $_POST['calon4'];
$namacalon5 = $_POST['calon5'];

```

```

if (strlen($namacalon1) > 1)
{
    $sqlinsert_infopeserta1 = mysql_query("insert
into `info_peserta` (nama_calon, id_list) values
('$namacalon1', '$idlist')");
    $sqlgetidcalon1 = mysql_query("SELECT
id_calon FROM `info_peserta` where nama_calon='$namacalon1'
and id_list='$idlist' ")
    or die (mysql_error());
    $dataidcalon1 =
mysql_fetch_array($sqlgetidcalon1);
    $idcalon1 = $dataidcalon1['id_calon'];

    $sqlinsert_calon1 = mysql_query("insert into
calon (id_calon, id_list, id_kota, nilai) values ('$idcalon1',
'$idlist', '$idkota', 0)");
}
.
.
.
.
$jumlahtps = $_POST['jumlahtps'];
$sqlinsert_tps = mysql_query("insert into tps
(id_propinsi, id_list, id_kota, jumlah) values ('$idprov',
'$idlist', '$idkota', '$jumlahtps')");
$insert4=1;
if ($insert1 == 1 && $insert2 == 1 && $insert3 == 1 &&
$insert4 == 1)
{ ?>
<script language="JavaScript">
    alert('Data Pemilu Kota/Kabupaten Berhasil
diinput!');
    document.location='admin.php';
</script>
<?php
}

```


Kode Sumber A. 11 Implementasi Mendaftar Petugas TPS

Action_inputpetugasbaru.php

```

$namapemilu = $_POST['namapemilu'];
$kotapemilu = $_POST['kotapemilu'];
$username = $_POST['username'];
$password = $_POST['password'];

$sqlgetidkota = mysql_query("SELECT * FROM kota where
nama_kota='$kotapemilu' ") or die (mysql_error());
$dataidkota = mysql_fetch_array($sqlgetidkota);
$idkota = $dataidkota['id_kota'];

$sqlgetidlist = mysql_query("SELECT * FROM `list
pemilu` where nama_pemilu='$namapemilu' ") or die
(mysql_error());
$dataidlist = mysql_fetch_array($sqlgetidlist);
$idlist = $dataidlist['id_list'];

$sqlcekjumlah = mysql_query("select * from petugas where
id_list='$idlist' and id_kota='$idkota' ") or die
(mysql_error());
$sqlambiljumlah = mysql_query("select * from tps
where id_list='$idlist' and id_kota='$idkota'") or die
(mysql_error());
$datajumlah = mysql_fetch_array($sqlambiljumlah);
$jumlahtps = $datajumlah['jumlah'];
if (mysql_num_rows($sqlcekjumlah) == $jumlahtps)
{
    ?>
    <script language="JavaScript">
        alert('Semua petugas di kota tersebut
telah diinput!');
        document.location='adminpetugasbaru.php';
    </script>
    <?php
}

```

```
else
{
    $sqlinsert_petugasbaru = mysql_query("insert
into petugas (id_list, id_kota, username, password,
flag_petugas)
        values ('$idlist', '$idkota',
'$username', '$password', 0)") or die (mysql_error());
    $insert1 = 1;
    if ($insert1 == 1)// && $insert2 == 1 &&
$insert3 == 1 && $insert4 == 1)
    { ?>
        <script language="JavaScript">
            alert('Data petugas TPS sampel
Berhasil diinput!');
        document.location='adminpetugasbaru.php';
        </script>
    }
    <?php
}
```

DAFTAR PUSTAKA

- [1] Aplikasi real count” [online] available : http://www.academia.edu/6417035/APLIKASI_REAL_COUNT_PEMILIHAN KEPALA DAERAH SEBAGAI ALAT UKUR PEMETAAN POLITIK PARTAI/. [diakses 12 Agustus 2015]
- [2] “Pemilihan umum di Indonesia” [online] available : https://id.wikipedia.org/wiki/Pemilihan_umum_di_Indonesia/. [diakses 12 Agustus 2015]
- [3] “Pengertian kepala daerah” [online] available : <http://artikelpengertianmakalah.blogspot.com/2015/05/pengertian-kepala-daerah-tugas-wewenang.html/>. [diakses 12 Agustus 2015]
- [4] “Pemilihan kepala daerah di Indonesia” [online] available : https://id.wikipedia.org/wiki/Pemilihan_kepala_daerah_di_Indonesia/. [diakses 12 Agustus 2015]
- [5] “Teknologi quick count dalam pemilihan umum” [online] available : <https://abdharisdreamer.wordpress.com/2014/08/13/teknologi-quick-count-dalam-pemilihan-umum/>. [diakses 7 Agustus 2015]
- [6] “Apa itu web service” [online] available : <http://hamdani.blog.ugm.ac.id/2011/07/15/apa-itu-web-service/> [diakses 16 Agustus 2015]
- [7] “Web service dan kegunaanya pada komputer” [online] available : <http://ketutrare.blogspot.com/2013/05/web-service-dan-kegunaanya-pada-sistem.html> [diakses 16 Agustus]
- [8] Dharma Kasman, Akhmad. 2013. Kolaborasi Dasyat Android dengan PHP & Mysql.
- [9] Andi. 2013. Wahana Komputer, Android Programming dengan Eclipse.

- [10] "Pengertian dan fungsi adobe dreamweaver" [online] available : <http://www.burung-net.com/2015/03/pengertian-dan-fungsi-adobe-dreamweaver.html> [diakses 1 Desember 2015]

BIODATA PENULIS



Erlangga Ibrahim Nur Wisisono, lahir di Nganjuk pada tanggal 14 Desember 1991, merupakan anak keempat dari sembilan bersaudara. Penulis telah menempuh pendidikan mulai SDN Pacewetan 1 (1997-2003), SMPN 1 Pace (2003-2006), SMAN 2 Nganjuk (2006-2009), dan terakhir sebagai mahasiswa Teknik Informatika ITS (2009-2016). Selama masa perkuliahan di Teknik Informatika ITS, penulis aktif dalam kegiatan keorganisasian, maupun komunitas.

Dalam menyelesaikan pendidikan sarjana, penulis mengambil bidang minat Rekayasa Perangkat Lunak (RPL). Penulis dapat dihubungi melalui email di erlangga.inw@gmail.com.